High-dimensional data: LMs and GLMs

Wei Pan

Division of Biostatistics and Health Data Science, School of Public Health,
University of Minnesota, Minneapolis, MN 55414
Email: panxx014@umn.edu

PubH 8475/Stat 8056

Linear Model and Least Squares

- Main ref: ESL §3.
- Data: (Y_i, X_i), $X_i = (X_{i1}, ..., X_{ip})'$, $i = 1, ..., n$.
  Y_i: continuous
- LM: $Y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i$,
  ϵ_i's iid with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$.
- $RSS(\beta) = \sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j)^2 = ||Y - X\beta||_2^2$.
- LSE (OLSE): $\hat{\beta} = \arg\min_{\beta} RSS(\beta) = (X'X)^{-1}X'Y$.
- Nice properties: Under true model,
  $E(\hat{\beta}) = \beta$,
  $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$,
  $\hat{\beta} \sim N(\beta, Var(\hat{\beta}))$,
  Gauss-Markov Theorem: $\hat{\beta}$ has min var among all linear unbiased estimates.

- Some questions:
  $\hat{\sigma}^2 = RSS(\hat{\beta})/(n - p - 1)$.
  Q: what happens if the denominator is n?
  Q: what happens if $X'X$ is (nearly) singular?
- What if p is large relative to n?
- Variable selection:
  forward, backward, stepwise: fast, but may miss good ones;
  best-subset: too time consuming.

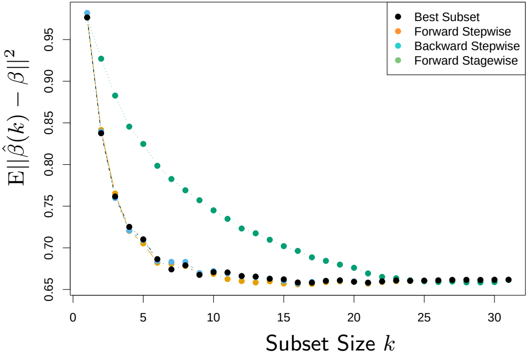


FIGURE 3.6. *Comparison of four subset-selection techniques on a simulated linear regression problem* $Y = X^T\beta + \varepsilon$. *There are* $N = 300$ *observations on* $p = 31$ *standard Gaussian variables, with pairwise correlations all equal to* 0.85. *For* 10 *of the variables, the coefficients are drawn at random from a* $N(0, 0.4)$ *distribution; the rest are zero. The noise*

Shrinkage or regularization methods

- Use regularized or penalized RSS:

  $$PRSS(\beta) = RSS(\beta) + \lambda J(\beta).$$

  λ: penalization parameter to be determined;
  (thinking about the p-value thresold in stepwise selection, or subset size in best-subset selection.)
  $J()$: prior; both a loose and a Bayesian interpretations; log prior density.
- Ridge: $J(\beta) = \sum_{j=1}^{p} \beta_j^2$; prior: $\beta_j \sim N(0, \tau^2)$.
  $\hat{\beta}^R = (X'X + \lambda I)^{-1} X'Y$.
- Properties: biased but small variances,
  $E(\hat{\beta}^R) = (X'X + \lambda I)^{-1} X'X\beta$,
  $Var(\hat{\beta}^R) = \sigma^2 (X'X + \lambda I)^{-1} X'X(X'X + \lambda I)^{-1} \leq Var(\hat{\beta})$,
  $df(\lambda) = tr[X(X'X + \lambda I)^{-1}X'] \leq df(0) = tr(X(X'X)^{-1}X') = tr((X'X)^{-1}X'X) = p$,

- Lasso: $J(\beta) = \sum_{j=1}^{p} |\beta_j|$.
  Prior: β_j Laplace or DE(0, τ^2);
  No closed form for $\hat{\beta}^L$.
- Properties: biased but small variances,
  $df(\hat{\beta}^L) = \#$ of non-zero $\hat{\beta}_j^L$'s (Zou et al).
- Special case: for $X'X = I$, or simple regression ($p = 1$),
  $\hat{\beta}_j^L = \text{ST}(\hat{\beta}_j, \lambda) = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \lambda)_+$,
  compared to:
  $\hat{\beta}_j^R = \hat{\beta}_j/(1+\lambda)$,
  $\hat{\beta}_j^H = \text{HT}(\hat{\beta}_j, \lambda) = \hat{\beta}_j I(|\hat{\beta}_j| > \lambda)$,
  $\hat{\beta}_j^B = \text{HT2}(\hat{\beta}_j, M) = \hat{\beta}_j I(\text{rank}(|\hat{\beta}_j)|) \leq M)$.
- A key property of Lasso: $\hat{\beta}_j^L = 0$ for large λ, but not $\hat{\beta}_j^R$.
  –simultaneous parameter estimation and selection.

- Note: for a convex $J(\beta)$ (as for Lasso and Ridge), min PRSS is equivalent to:
  min $RSS(\beta)$ s.t. $J(\beta) \leq t$.
- Offer an intutive explanation on why we can have $\hat{\beta}_j^L = 0$; see Fig 3.11.
  Theory: $|\beta_j|$ is singular at 0; Fan and Li (2001).
- How to choose λ?
  obtain a solution path $\hat{\beta}(\lambda)$, then, as before, use tuning data or CV or model selection criterion (e.g. AIC or BIC).
- Least Angle Regression (LARS): fast to find solution paths in LMs.
- Example: R code ex3.1.r

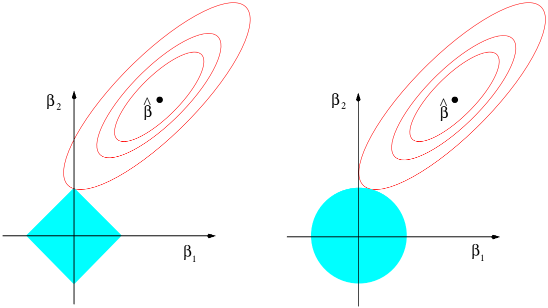
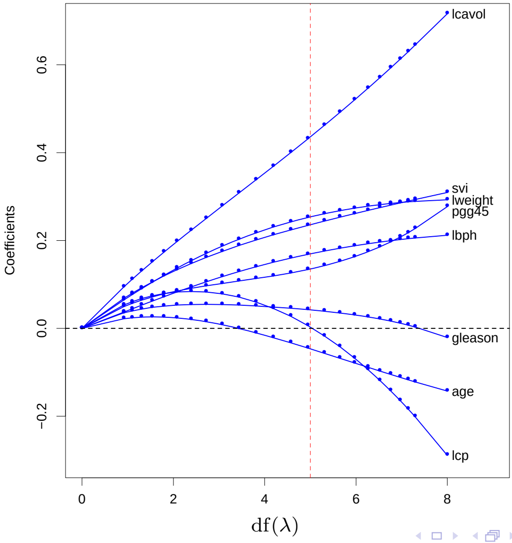
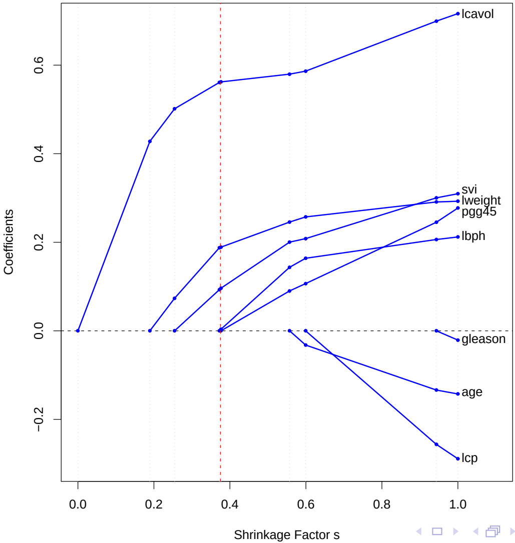


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \le t$ and $\beta_1^2 + \beta_2^2 \le t^2$, respectively, while the red ellipses are the contours of the least squares error function.*


lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
Coefficients
−0.2
0.0
0.2
0.4
0.6
0
2
4
6
8
df(λ)


lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
Coefficients
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Shrinkage Factor s

- Lasso: biased estimates; alternatives:
- Relaxed lasso: 1) use Lasso for VS; 2) then use LSE or MLE on the selected model.
- Use a non-convex penalty:
  SCAD: eq (3.82) on p.92;
  Bridge $J(\beta) = \sum_j |\beta_j|^q$ with $0 < q < 1$;
  Adaptive Lasso (Zou 2006): $J(\beta) = \sum_j |\beta_j|/|\tilde{\beta}_{j,0}|$;
  Truncated Lasso Penalty (Shen, Pan &Zhu 2012, JASA):
  $\text{TLP}(\beta; \tau) = \sum_j \min(|\beta_j|, \tau)$, or
  $\text{TLP}(\beta; \tau) = \sum_j \min(|\beta_j|/\tau, 1) \to I(\beta \neq 0)$ as $\tau \to 0^+$.
  MCP: ...
- Choice b/w Lasso and Ridge: bet on a sparse model?
  risk prediction for GWAS (Austin, Pan & Shen 2013, *SADM*).
- Elastic net (Zou & Hastie 2005):

$$J(\beta) = \sum_j \alpha|\beta_j| + (1-\alpha)\beta_j^2$$

  may select more (correlated) X_j's.

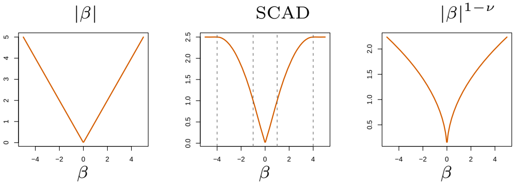


FIGURE 3.20. *The lasso and two alternative non–convex penalties designed to penalize large coefficients less. For SCAD we use $\lambda = 1$ and $a = 4$, and $\nu = \frac{1}{2}$ in the last panel.*

- **Group** Lasso: a group of variables $\beta_{(g)} = (\beta_{j1}, ..., \beta_{jp_g})'$ are to be 0 (or not) at the same time,

  $$J(\beta) = \sum_g \sqrt{p_g} ||\beta_{(g)}||_2$$

  L_2-norm; not L_1/Lasso or **squared** L_2/Ridge.
  better in VS (but worse for parameter estimation?)
- Group SCAD: $J(\beta) = \sum_g \sqrt{p_g} \text{SCAD}(||\beta_{(g)}||_2)$
- Group TLP: $J(\beta, \tau) = \sum_g \sqrt{p_g} \text{TLP}(||\beta_{(g)}||_2; \tau)$
- Sparse Group Lasso: $J(\beta) = (1 - \alpha) \sum_g \sqrt{p_g} ||\beta_{(g)}||_2 + \alpha ||\beta||_1$
- **Grouping**/fusion penalties: encouraging equalities b/w β_j's (or $|\beta_j|$'s).
  - Fused Lasso: $J(\beta) = \sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|$
    $J(\beta) = \sum_{(j,k) \in G} |\beta_j - \beta_k|$
  - Generalized Lasso: $J(\beta) = ||D\beta||_1$
  - Grouping pursuit (Shen & Huang 2010, JASA):

    $$J(\beta; \tau) = \sum_{j=1}^{p-1} TLP(\beta_j - \beta_{j+1}; \tau)$$

- Grouping penalties:
    - Zhu, Shen & Pan (2013, JASA):

$$J_2(\beta;\tau) = \sum_{j=1}^{p-1} TLP(|\beta_j| - |\beta_{j+1}|;\tau);$$

$$J(\beta;\tau_1,\tau_2) = \sum_{j=1}^{p} TLP(\beta_j;\tau_1) + J_2(\beta;\tau_2);$$

    - Kim, Pan & Shen (2013, Biometrics):

$$J_2'(\beta) = \sum_{j\sim k} \left| I(\beta_j \neq 0) - I(\beta_k \neq 0) \right|;$$

$$J_2(\beta;\tau) = \sum_{j\sim k} \left| TLP(\beta_j;\tau) - TLP(\beta_k;\tau) \right|;$$

- Dantzig Selector (§3.8).
- Theory (§3.8.5).

Logistic regression

- Binary or multinomial logit model: for $k = 1, ..., K - 1$,

  $$\log \frac{Pr(k|x)}{Pr(K|x)} = \beta_{0,k} + x'\beta_{1,k},$$

  or equivalently,

  $$Pr(k|x) = \frac{\exp(\beta_{0,k} + x'\beta_{1,k})}{1 + \sum_{l=1}^{K-1} \exp(\beta_{0,k} + x'\beta_{1,k})}.$$

  Then $\hat{G}(x) = \arg\max_k Pr(k|x)$.
- x can be expanded to include high-order terms.
- Parameter estimation: MLE
  Note: approx equivalent to fitting multiple binary logit models separetely (Begg & Gray, 1984, Biometrika).
- Logistic reg vs L/QDA: the former is more general; the latter has a stronger assumption and thus possibly more efficient if ...; Logistic reg is quite good.
- Example code: ex4.1.r

Penalized logistic regression (§18.3.2, 18.4)

- Need VS or regularization for a large p.
- Add a penalty term $J(\beta)$ to $-\log L$
  $J(\beta)$ can be Lasso, ..., as before.
- Computing algorithms: a Taylor expansion (i.e. quadratic approx) of $\log L$, then the same as penalized LR.
- R package `glmnet`: an elastic net penalty.
  hence do either Lasso or Ridge (or both).

R packages for penalized GLMs (and Cox PHM)

- `glmnet`: Ridge, Lasso and Elastic net.
- `ncvreg`: SCAD, MCP.
- `glmtlp`: TLP.
- `grpreg`: group Lasso, group SCAD, ...
- `seagull`, `SGL`: sparse group Lasso.
- `genlasso`: generalized Lasso for LMs, including fused Lasso.
- `FGSG`: grouping/fusion penalties (based on Lasso, TLP, etc) for LMs
- More general convex programming: `CVXR`; like CVX, CVXPY.
- Example 3.3.R

Computational Algorithms

- Quadratic programming: the original for Lasso; slow.
- LARS (§3.8): the solution path is piece-wise linear; at a cost of fitting several single LMs; not general?
- Incremental Forward Stagewise Regression (§3.8): approx; related to boosting.
- A simple (and general) way: $|\beta_j| = \beta_j^2/|\hat{\beta}_j^{(r)}|$;
  truncate a current estimate $|\hat{\beta}_j^{(r)}| \approx 0$ at a small ϵ.
- Coordinate-descent algorithm (§3.8.6): update each β_j while fixing others at the current estimates–recall we have a closed-form solution for a single β_j!
  simple and general but not applicable to grouping penalties.
- ADMM (Boyd et al 2011).
  `http://stanford.edu/~boyd/admm.html`
- For TLP: iterating b/w Difference of Convex (DC) (or MM alg.) and (weighted) lasso

Inference

- Q: How to get a p-value or CI for a predictor?
  Challenges: biased estimates; selection bias
- Sample splitting (to two parts): 1. using the training data for (Lasso) penalized reg (for VS); 2. using the validation data to fit the selected model for inference by OLSE or MLE.
  Refs: Wasserman & Roeder (2009, AoS); Meinshausen, Meier & Bühlmann (2009, JASA).
  +: simple; more general.
  -: loss of efficiency. Better with repeated/multiple splitting.
  R package: `hdi`, function `multi.split()` or `hdi()`.
- Debiased/de-sparsified lasso (or lasso projection): next page.
  R package: `hdi`, function `lasso.proj()`.
- Ref: Dezeure et al (2015, *Stat Sci*).
  `https://arxiv.org/pdf/1408.4026.pdf`
  Example: ex3.4.R

Lasso projection

- Model: $Y = X\beta + \epsilon$, $X = (X^{(1)}, X^{(2)}, ..., X^{(p)})$
- Fact 1: $\beta_j \neq b_j$ unless ...
  working model: $Y = X^{(j)} b_j + e$
- Fact 2: LSEs $\hat{\beta}_j = \hat{b}_j$ if $p < n$ AND
  $Y = Z^{(j)} b_j + e$, $Z^{(j)}$ is a residual vector of regressing $X^{(j)}$ on all other $X^{(k)}$'s with $k \neq j$.
  Why? $Z^{(j)} \perp X^{(k)}$
  $\hat{b}_j = (Z^{(j)})' Y / (Z^{(j)})' Z^{(j)} = (Z^{(j)})' Y / (Z^{(j)})' X^{(j)}$.
  $E(\hat{b}_j) = \beta_j + \sum_{k \neq j} P_{jk} \beta_k$, $P_{jk} = (Z^{(j)})' X^{(k)} / (Z^{(j)})' X^{(j)}$.
  $P_{jk} = 0$.
- For $p > n$, use Lasso to get $Z^{(j)}$, then $P_{jk} \neq 0$.
  $\hat{\beta}_{C,j} = \hat{b}_j - \sum_{k \neq j} P_{jk} \hat{\beta}_k$,
  $\hat{\beta}$: Lasso estimates.
  $\hat{\beta}_{C,j} \sim N(\beta_j, v_j)$.

Inference

- TLP/SCAD: if interested in β_j (that can be high-d for TLP), 1. use the whole sample to fit a penalized reg model by penalizing all parameters **except** β_j; 2. apply the usual Wald or LRT to get the p-value or CI for β_j.
  Refs: Zhu, Shen & Pan (2020, JASA); Shi et al (2019, AoS).
- Model-X Knockoffs: FDR control for VS.
  R package: `knockoff`.
  `https://web.stanford.edu/group/candes/knockoffs/index.html`
- Conformal inference: can give prediction intervals; ...
  R package: `https://github.com/ryantibs/conformal`

Sure Independence Screening (SIS)

- Q: penalized (or stepwise ...) regression can do automatic VS; just do it?
- Key: there is a cost/limit in performance/speed/theory.
- Q2: some methods (e.g. LDA/QDA/RDA) do not have VS, then what?
- Going back to basics: first conduct VS in marginal analysis,
  1) $Y \sim X_1$, $Y \sim X_2$, ..., $Y \sim X_p$;
  2) choose a few top ones, say p_1;
  p_1 can be chosen somewhat arbitrarily, or treated as a tuning parameter
  3) then apply penalized reg (or other VS) to the selected p_1 variables.
- Called SIS with theory (Fan & Lv, 2008, JRSS-B).
  R package `SIS`;
  iterative SIS (ISIS); why? a limitation of SIS ...

Using Derived Input Directions

- PCR: PCA on X, then use the first few PCs as predictors.
  Use a few top PCs explaining a majority (e.g. 85% or 95%) of total variance;
  # of components: a tuning parameter; use (genuine) CV;
  Used in genetic association studies, even for $p < n$ to improve power.
  $+$: simple;
  -: PCs may not be related to Y.

- Partial least squares (PLS): multiple versions; see Alg 3.3.
  Main idea:
  1) regress Y on each X_j univariately to obtain coef est ϕ_{1j};
  2) first component is $Z_1 = \sum_j \phi_{1j} X_j$;
  3) regress X_j on Z_1 and use the residuals as new X_j;
  4) repeat the above process to obtain Z_2, ...;
  5) Regress Y on Z_1, Z_2, ...
- Choice of # components: tuning data or CV (or AIC/BIC?)
- Contrast PCR and PLS:
  PCA: $\max_\alpha \text{Var}(X\alpha)$ s.t. ....;
  PLS: $\max_\alpha \text{Cov}(Y, X\alpha)$ s.t. ...;
  Continuum regression (Stone & Brooks 1990, JRSS-B)
- Penalized PCA (...) and Penalized PLS (Huang et al 2004, BI; Chun & Keles 2012, JRSS-B; R packages ppls, spls).
- Example code: ex3.2.r

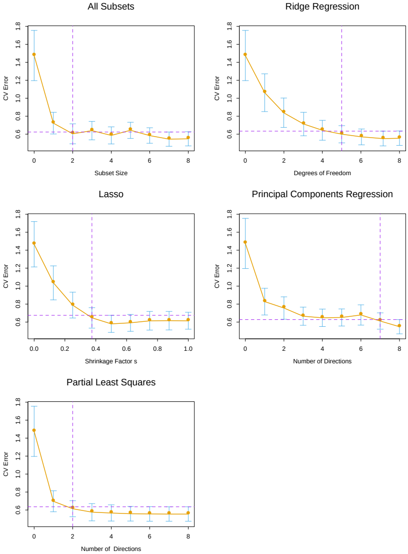


FIGURE 3.7. *Estimated prediction error curves and their standard errors for the various selection and*