

On Algorithms for Solving Least Squares Problems under an L_1 Type Penalty or Constraint

Berwin A Turlach

berwin@maths.uwa.edu.au

School of Mathematics and Statistics (M019)

The University of Western Australia

35 Stirling Highway

Crawley, WA 6009

Australia

Characterisation of solutions

If $\hat{\beta}$ is a solution of (2) if $\lambda \geq 0$ exists such that

$$\mathbf{X}^T \hat{\mathbf{r}} = \lambda \mathbf{c},$$

where $\hat{\mathbf{r}} = \mathbf{y} - \mathbf{X}^T \hat{\beta}$ and $\mathbf{c} = (c_1, \dots, c_m)^T$ is such that

$$c_i \begin{cases} = 1 & \text{if } \hat{\beta}_i > 0 \\ = -1 & \text{if } \hat{\beta}_i < 0 \\ \in [-1, 1] & \text{if } \hat{\beta}_i = 0 \end{cases}$$

Note that $\|\mathbf{c}\|_\infty = 1$ and $\mathbf{c}^T \hat{\beta} = \|\hat{\beta}\|_1$.

Hence

$$\lambda = \|\mathbf{X}^T \hat{\mathbf{r}}\|_\infty = \hat{\mathbf{r}}^T \mathbf{X} \hat{\beta} / \|\hat{\beta}\|_1$$

Osborne *et al.* (2000a,b); Efron *et al.* (2004)

The LASSO

$$\text{minimise}_{\beta_1, \dots, \beta_m} \frac{1}{2} \sum_{i=1}^n \left(y_i - \sum_{j=1}^m x_{ij} \beta_j \right)^2 \quad (1a)$$

$$\text{subject to} \quad \sum_{j=1}^m |\beta_j| \leq t, \quad (1b)$$

or, equivalently,

$$\text{minimise}_{\beta \in \mathbb{R}^m} f(\beta) = \frac{1}{2} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \quad (2a)$$

$$\text{subject to} \quad g(\beta) = t - \|\beta\|_1 \geq 0. \quad (2b)$$

Tibshirani (1996)

An algorithm to calculate all solutions

1. (a) Set $\hat{\mu}^{(0)} = \mathbf{0}$, $\hat{\beta}^{(0)} = \mathbf{0}$ and $k = 0$.
 (b) Calculate $\mathbf{c} = X^T \mathbf{y}$ and set $C = \|\mathbf{c}\|_\infty = \max_j \{ |c_j| \}$.
 (c) Let $\sigma = \{j : |c_j| = C\}$.
2. (a) Set $X_\sigma = (\dots x_j \dots)_{j \in \sigma}$ and calculate

$$\bar{\mathbf{b}}_\sigma^{(k+1)} = (X_\sigma^T X_\sigma)^{-1} X_\sigma^T \mathbf{y}$$

$$\bar{\mu}^{(k+1)} = X_\sigma \bar{\mathbf{b}}_\sigma^{(k+1)}$$

- (b) And use these results to update:

$$\hat{\mu}^{(k+1)} = \hat{\mu}^{(k)} + \gamma \left(\bar{\mu}^{(k+1)} - \hat{\mu}^{(k)} \right)$$

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + \gamma \left(\bar{\beta}^{(k+1)} - \hat{\beta}^{(k)} \right)$$

where $\bar{\beta}^{(k+1)} = \mathbf{P} \begin{pmatrix} \bar{\mathbf{b}}_\sigma^{(k+1)} \\ \mathbf{0} \end{pmatrix}$; \mathbf{P} a suitable permutation matrix.

An algorithm to calculate all solutions

2. (c) The step size γ is $\gamma = \min(\gamma_1, \gamma_2, 1)$ where, with

$$\gamma_1 = \min_{j \notin \sigma}^+ \left\{ \frac{C - c_j}{C - a_j}, \frac{C + c_j}{C + a_j} \right\}, \quad \gamma_2 = \min_{j \in \sigma}^+ \left\{ -\frac{\hat{\beta}_j^{(k)}}{\hat{\beta}_j^{(k+1)} - \hat{\beta}_j^{(k)}} \right\}$$

and $\mathbf{a} = X^T (\bar{\mu}^{(k+1)} - \hat{\mu}_k)$.

(d) If $\gamma = \gamma_1$, then

$$\sigma = \sigma \cup \left\{ j : j \notin \sigma \text{ and } \gamma_1 = \min^+ \left\{ \frac{C - c_j}{C - a_j}, \frac{C + c_j}{C + a_j} \right\} \right\}$$

If $\gamma = \gamma_2$, then $\sigma = \sigma \setminus \{j : \hat{\beta}_j^{(k+1)} = 0\}$.

(e) Calculate $\mathbf{c} = X^T (\mathbf{y} - \hat{\mu}^{(k+1)})$, set $C = \max_j \{|c_j|\}$ and

$k \leftarrow k + 1$.

If $C = 0$ stop, otherwise return to step 2a

Osborne *et al.* (2000a), Efron *et al.* (2004)

An algorithm for fixed t

Osborne *et al.* (2000b) proposed the following algorithm which is based on a local linearisation of (2a) about a current β .

At each step solve the following optimisation problem:

$$\underset{\mathbf{h}}{\text{minimise}} \quad f(\beta + \mathbf{h}) \quad (4a)$$

$$\text{where} \quad \theta_\sigma^T (\beta_\sigma + \mathbf{h}_\sigma) \leq t \quad \text{and} \quad \mathbf{h} = \mathbf{P} \begin{pmatrix} \mathbf{h}_\sigma \\ \mathbf{0} \end{pmatrix} \quad (4b)$$

β_i may be non-zero if and only if $i \in \sigma$.

$\theta_\sigma = \text{sign}(\beta_\sigma)$.

At any step of the algorithm β_σ has to be feasible ($\theta_\sigma^T \beta_\sigma \leq t$).

\mathbf{P} is the permutation matrix such that $\beta = \mathbf{P} \begin{pmatrix} \beta_\sigma \\ \mathbf{0} \end{pmatrix}$.

An algorithm for fixed t

Let $\tilde{\beta} = \beta + \mathbf{h}$ be the solution of (4).

If $\text{sign}(\tilde{\beta}_\sigma) = \theta_\sigma$ then we call $\tilde{\beta}$ *sign feasible*.

If $\tilde{\beta}$ is not sign feasible, we proceed as follows:

1. Move to the first new zero component in direction \mathbf{h} , i.e. find the smallest γ , $0 < \gamma < 1$ and corresponding $k \in \sigma$ such that $0 = \beta_k + \gamma h_k$.
2. Update σ by deleting k from it, setting $\beta = \beta + \gamma \mathbf{h}$, resetting β_σ and θ_σ accordingly (they are still both feasible) and recompute \mathbf{h} by solving (4) again.
3. Iterate until a sign feasible $\tilde{\beta}$ is obtained.

An algorithm for fixed t

If $\tilde{\beta}$ is sign feasible, then we can test it for optimality.

Calculate

$$\tilde{\mathbf{v}} = \mathbf{X}^T \tilde{\mathbf{r}} / \|\mathbf{X}^T \tilde{\mathbf{r}}\|_\infty$$

If $|\tilde{v}_i| = 1$ for $i \in \sigma$ and $-1 \leq \tilde{v}_i \leq 1$ for $i \notin \sigma$, then $\tilde{\beta}$ is a solution of (2). Otherwise, we proceed as follows.

1. Determine the most violated condition, i.e. find s such that \tilde{v}_s has maximal absolute value.
2. Update σ by adding s to it. β_σ and θ_σ are updated by appending a zero and $\text{sign}(\tilde{v}_s)$, respectively, as last elements.
3. Solve (4) and iterate.

Constrained vs. penalised estimation

The constrained problem (2) is, of course, equivalent to the penalised problem:

$$\underset{\beta \in \mathbb{R}^m}{\text{minimise}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_1 \quad (5)$$

There seems to be no algorithm that solves (5) directly for given λ . In particular in the case where $n < m$.

From the characterisation of solutions of (2) it follows that if $\lambda \geq \|\mathbf{X}^T \mathbf{y}\|_\infty$, then the solution to (5) is $\hat{\beta} = \mathbf{0}$.

An algorithm for fixed λ

Use the same ideas as for the fixed t case.

At each step solve the following optimisation problem:

$$\underset{\mathbf{h}}{\text{minimise}} f(\beta + \mathbf{h}) + \lambda \theta_\sigma^T (\beta_\sigma + \mathbf{h}_\sigma) \quad (6)$$

β_i may be non-zero if and only if $i \in \sigma$.

$\theta_\sigma = \text{sign}(\beta_\sigma)$.

\mathbf{P} is the permutation matrix such that $\beta = \mathbf{P} \begin{pmatrix} \beta_\sigma \\ \mathbf{0} \end{pmatrix}$.

And $\mathbf{h} = \mathbf{P} \begin{pmatrix} \mathbf{h}_\sigma \\ \mathbf{0} \end{pmatrix}$.

An algorithm for fixed λ

Let $\tilde{\beta} = \beta + \mathbf{h}$ be the solution of (6).

If $\text{sign}(\tilde{\beta}_\sigma) = \theta_\sigma$ then we call $\tilde{\beta}$ *sign feasible*.

If $\tilde{\beta}$ is not sign feasible, we proceed as follows:

1. Move to the first new zero component in direction \mathbf{h} , i.e. find the smallest γ , $0 < \gamma < 1$ and corresponding $k \in \sigma$ such that $0 = \beta_k + \gamma h_k$.
2. Update σ by deleting k from it, setting $\beta = \beta + \gamma \mathbf{h}$, resetting β_σ and θ_σ accordingly and recompute \mathbf{h} by solving (6) again.
3. Iterate until a sign feasible $\tilde{\beta}$ is obtained.

An algorithm for fixed λ

If $\tilde{\beta}$ is sign feasible, then we can test it for optimality.

Calculate

$$\tilde{\mathbf{c}} = \mathbf{X}^T \tilde{\mathbf{r}}$$

If $\tilde{c}_i = \text{sign}(\tilde{\beta}_i) \lambda$ for $i \in \sigma$ and $-\lambda \leq \tilde{c}_i \leq \lambda$ for $i \notin \sigma$, then $\tilde{\beta}$ is a solution of (5). Otherwise, we proceed as follows.

1. Determine the most violated condition, i.e. find s such that \tilde{c}_s has maximal absolute value.
2. Update σ by adding s to it. β_σ and θ_σ are updated by appending a zero and $\text{sign}(\tilde{c}_s)$, respectively, as last elements.
3. Solve (6) and iterate.

References

- Chen, S.S., Donoho, D.L. and Saunders, M.A. (1999). Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing* **20**(1): 33–61.
URL: <http://www-stat.stanford.edu/~donoho/Reports/1995/30401.ps.Z>
- Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion), *Annals of Statistics* **32**(2): 407–499.
- Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**(456): 1348–1360.
- Fu, W.J. (1998). Penalized regression: The Bridge versus the Lasso, *Journal of Computational and Graphical Statistics* **7**(3): 397–416.
- Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization, in L. Niklasson, M. Bodén and T. Ziemse (eds), *ICANN '98, Perspectives in Neural Computing*, Vol. 1, Springer-Verlag, pp. 201–206.
URL: <http://www.hds.utc.fr/~grandval/>
- Grandvalet, Y. and Canu, S. (1999). Outcomes of the equivalence of adaptive ridge with least absolute shrinkage, in M. Kearns, S. Solla and D. Cohn (eds), *NIPS'1998*, Vol. 11, MIT Press, pp. 445–451.
URL: <http://www.hds.utc.fr/~grandval/>
- Lokhorst, J. (1999). *The LASSO and Generalised Linear Models*, Honours project, Department of Statistics, The University of Adelaide, South Australia, Australia.
- Osborne, M.R., Presnell, B. and Turlach, B.A. (2000a). A new approach to variable selection in least squares problems, *IMA Journal of Numerical Analysis* **20**(3): 389–403.

13-1

Concluding remarks

- The presented algorithms are specifically designed to solve (2) or (5) and are quite efficient in doing so.
- They can be easily adapted to problems where the quadratic loss function is replaced by another loss function by embedding them within an IRLS loop (Lokhorst, 1999; Roth, 2002, 2004).
- Other algorithms have been proposed
 - in the wavelet literature by Chen *et al.* (1999) and Sardy *et al.* (2000);
 - as special cases of generalisations that allow either a more general penalty or a more general loss function (Fu, 1998; Fan and Li, 2001);
 - as an adaptive ridge regression procedure (Grandvalet, 1998; Grandvalet and Canu, 1999).

LASSO algorithms

JSM 2004, Toronto – p. 13

- Osborne, M.R., Presnell, B. and Turlach, B.A. (2000b). On the LASSO and its dual, *Journal of Computational and Graphical Statistics* **9**(2): 319–337.
- Roth, V. (2002). The Generalized LASSO: a wrapper approach to gene selection for microarray data, *Technical Report IAI-TR-2002-8*, Institute of Computer Science III, University of Bonn, Germany.
URL: <http://www2.inf.ethz.ch/~vroth/>
- Roth, V. (2004). The generalized LASSO, *IEEE Transactions on Neural Networks* **15**(1): 16–28.
- Sardy, S., Bruce, A.G. and Tseng, P. (2000). Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries, *Journal of Computational and Graphical Statistics* **9**(2): 361–379.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B* **58**(1): 267–288.

13-2