STRONG RULES FOR EFFICIENT LASSO COMPUTATIONS AND THE BASIL ALGORITHM

NOTES BY PARKER KNIGHT

1. Strong rules for the LASSO

1.1. **Subgradients.** Let $f : \mathbb{R}^m \to \mathbb{R}$ be convex [1].

Recall the following first order condition: if f is differentiable, then

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall x, y \in \text{dom}(f)$$

What if f is not differentiable? This motivates the following definition: Call $g \in \mathbb{R}^m$ a *subgradient* of f at x iff

$$f(y) \geq f(x) + g^T (y - x) \quad \forall x, y \in \text{dom}(f)$$

The subdifferential of f at x, denote $\partial f(x)$, is the set of all subgradients. The following facts will be useful:

(1) If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$
(2) For $\alpha_1, \alpha_2 \geq 0$, then $\partial \left[\alpha_1 f_1(x) + \alpha_2 f_2(x)\right] = \alpha_1 \partial f_1(x) + \alpha_2 \partial f_2(x)$
(3) x^* minimizes f iff $0 \in \partial f(x^*)$

where we define set addition as $A + B = \{a + b | a \in A, b \in B\}$.

1.2. **The LASSO.** Recall the LASSO [2] loss function:

$$Q_\lambda(\beta) = \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

where the LASSO solution $\hat{\beta}(\lambda)$ satisfies

$$Q_\lambda(\hat{\beta}(\lambda)) = \min_\beta Q_\lambda(\beta)$$

Note that $Q_\lambda(.)$ is not differentiable everywhere, but it is convex. So $\hat{\beta}_\lambda$ minimizes Q iff $0 \in \partial Q_\lambda(\hat{\beta}(\lambda))$.

But how do we find ∂Q?

$$\partial Q_\lambda(b) = -X^T (Y - Xb) + \lambda\gamma$$

where $\gamma \in \partial ||b||_1$. For $g(x) = |x|$, we have

$$\partial g(x) = \begin{cases} \{1\} & x > 0 \\ \{-1\} & x < 0 \\ [-1, 1] & x = 0 \end{cases}$$

since g is differentiable when $x \neq 0$, and when $x = 0$ we have $|y| \geq \alpha y$ iff $\alpha \in [-1, 1]$. We extend this component-wise[1] to get the expression for γ:

$$\gamma_j \in \begin{cases} \{1\} & b_j > 0 \\ \{-1\} & b_j < 0 \\ [-1, 1] & b_j = 0 \end{cases}$$

for $j = 1, ..., p$. Using this, we have that $\hat{\beta}(\lambda)$ minimizes Q iff

$$X_j^T(Y - X\hat{\beta}(\lambda)) = \lambda\gamma_j$$

for $j = 1, ..., p$ with γ_j defined as above for $\hat{\beta}(\lambda)$. This admits a key detail: $|X_j^T(Y - X\hat{\beta}(\lambda))| < \lambda$ implies $\hat{\beta}_j(\lambda) = 0$.

1.3. **Strong rules.** The material in this section comes largely from [2].

Suppose we have a grid of tuning parameters $\lambda_1, ..., \lambda_L$, which which we seek exact LASSO solutions. The optimality condition described above can lead us to an algorithm which will return exact solutions, but only needs to solve a series of smaller sub-problems.

First we need an additional assumption. Let $c_j(\lambda_k) = X_j^T(Y - X\hat{\beta}(\lambda_k))$ for $j = 1, ..., p$.

Assumption 1. $|c_j(\lambda) - c_j(\tilde{\lambda})| \leq |\lambda - \tilde{\lambda}| \quad \forall \lambda, \tilde{\lambda} > 0$

This says that $c_j(.)$ is non-expansive in its argument.

Theorem 1. *Suppose* $|c_j(\lambda_{k-1})| < 2\lambda_k - \lambda_{k-1}$ *and Assumption 1 holds. Then* $\hat{\beta}_j(\lambda_k) = 0$.

Proof. Observe

$$\begin{aligned} |c_j(\lambda_k)| &= |c_j(\lambda_k) - c_j(\lambda k - 1) + c_j(\lambda_{k-1})| \\ &\leq |c_j(\lambda_k) - c_j(\lambda k - 1)| + |c_j(\lambda_{k-1})| \\ &< (\lambda_{k-1} - \lambda_k) + (2\lambda_k - \lambda_{k-1}) \\ &= \lambda_k \end{aligned}$$

By the optimality condition for the LASSO problem this implies $\hat{\beta}_j(\lambda_k) = 0$. □

Theorem 1 yields a natural algorithm for computing the LASSO solution for a grid of tuning parameters.

1.3.1. *The strong rules algorithm.* This method requires data X, outcome Y, tuning parameters $\lambda_1 ... \lambda_L$, and an initial estimate $\hat{\beta}(\lambda_1)$[2] Then, for $k = 1, ..., L - 1$:

I Let $\upsilon \subset \{1, ..., p\}$ denote the set of eligible predictors, and let $S(\lambda_k) \subset \{1, ..., p\} = \{j : |c_j(\lambda_k)| >= 2\lambda_{k+1} - \lambda_k\}$. Set $\upsilon = S(\lambda_k)$.

II Solve the LASSO problem using only the predictors in υ.

III Check the subgradient optimality condition at *all* predictors in X. If none of them violate the condition, we are done, yielding $\hat{\beta}(\lambda_{k+1})$. If any of them violate the condition, add these predictors to υ and repeat steps two and three.

[1] Proof is simple, but requires a bit more subgradient calculus to do formally.

[2] This initial estimate could correspond to an intercept-only model, for instance.

2. BASIL

The Batch Screening Iterative LASSO (BASIL) [3] algorithm takes advantage of the strong rules method to yield a correct LASSO solution for a set of tuning parameters while minimizing IO operations. This is valuable for massive genetic data, where the number of predictors may be too large to load into memory, and IO disk operations become expensive.

2.1. **The Algorithm.** The BASIL algorithm is comprised of three main phases: screening (removing variables according to the strong rules), fitting (compute the LASSO fit with a subset of the variables), and checking (ensuring that optimality conditions are satisfied at all covariates).

The algorithm requires the following inputs:

- data $X, Y, \Omega = \{1, ..., p\}$
- initial residual $r^{(0)}$ and active set $A^{(0)} = \emptyset$.
- integers $M, \Delta M, \ell, \ell'$
- master tuning parameter list $\Lambda = \{\lambda_1, ..., \lambda_L\}$ and initial list $\Lambda^{(0)} = \{\lambda_1, ..., \lambda_\ell\}$

Then, at iterate k:

2.1.1. *Screening.* For each $j \in \Omega/A^{(k)}$, compute $c_j^{(k)} = X_j^T r^{(k)}$. Let $\upsilon_M^{(k)}$ denote the set of indices in $\Omega/A^{(k)}$ with the M largest values of $c_j^{(k)}$. Define the strong set $S^{(k)} = A^{(k)} \cup \upsilon_M^{(k)}$. The key assumption of this step: among the variables that aren't already known to be active, those with the M largest c values are likely to be active (pass the strong rules screening) at this iterate.

2.1.2. *Fitting.* For each $\lambda \in \Lambda^{(k)}$ fit the LASSO using only the covariates in $S^{(k)}$. Save $\hat{\beta}^{(k)}(\lambda)$ and residual $r^{(k)}(\lambda)$.

2.1.3. *Checking.* Find the smallest $\lambda \in \Lambda^{(k)}$ at which the KKT optimality condition is satisfied:

$$\bar{\lambda}^{(k)} = \min \left\{ \lambda \in \Lambda^{(k)} : \max_{j \in \Omega/S^{(k)}} \left[\frac{1}{n} |X_j^T r^{(k)}(\lambda)| \right] < \lambda \right\}$$

If this set is empty, increase M by ΔM and repeat the Screening and Fitting steps (this wil increase the size of the strong set). Otherwise, let $r^{(k+1)} = r^{(k)}(\bar{\lambda}^{(k)})$, set $A^{(k+1)}$ to be the active set at $\hat{\beta}^{(k)}(\bar{\lambda}^{(k)})$, and let $\Lambda^{(k+1)} = \{\lambda \in \Lambda^{(k)} : \lambda < \bar{\lambda}^{(k)}\} \cup \{\text{the next } \ell' \text{ tuning parameters in } \Lambda\}$.

For $\lambda \in \Lambda^{(k)}/\Lambda^{(k+1)}$, we found exact LASSO solutions. Continue iterating until we have solutions for all $\lambda \in \Lambda$.

2.2. **Discussion.** The BASIL algorithm improves performance through its screening step. The inner products c_j can be easily computed in parallel at each iteration. Assuming that the strong set is approximately correct for moderately-sized subsets of tuning parameters, this allows us to load only a small subset of variables into memory before fitting the model for multiple tuning parameters at once. Issues may arise with very small λ values, since these are likely to have larger active sets - if M is not sufficiently large to capture all of the active variables, this could lead to a large number of iterations being needed to fit the correct model. However, making M too large defeats the very purpose of the screening step, so care must

be taken. The authors of [3] discuss this, and provide a principled heuristic for choosing an appropriate M.

References

[1] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press.
[2] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani, "Strong rules for discarding predictors in lasso-type problems: Strong rules for discarding predictors," vol. 74, no. 2, pp. 245–266.
[3] J. Qian, Y. Tanigawa, W. Du, M. Aguirre, C. Chang, R. Tibshirani, M. A. Rivas, and T. Hastie, "A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK biobank," vol. 16, no. 10, p. e1009141.