NATURAL COORDINATE DESCENT ALGORITHM FOR L1-PENALISED REGRESSION IN GENERALISED LINEAR MODELS

TOM MICHOEL

ABSTRACT. The problem of finding the maximum likelihood estimates for the regression coefficients in generalised linear models with an $\ell_1$ sparsity penalty is shown to be equivalent to minimising the unpenalised maximum log-likelihood function over a box with boundary defined by the $\ell_1$-penalty parameter. In one-parameter models or when a single coefficient is estimated at a time, this result implies a generic soft-thresholding mechanism which leads to a novel coordinate descent algorithm for generalised linear models that is entirely described in terms of the natural formulation of the model and is guaranteed to converge to the true optimum. A prototype implementation for logistic regression tested on two large-scale cancer gene expression datasets shows that this algorithm is efficient, particularly so when a solution is computed at set values of the $\ell_1$-penalty parameter as opposed to along a regularisation path. Source code and test data are available from http://glmnat.googlecode.com.

1. INTRODUCTION

In high-dimensional regression problems where the number of potential model parameters greatly exceeds the number of training samples, the use of an $\ell_1$ penalty which augments standard objective functions with a term that sums the absolute effect sizes of all parameters in the model has emerged as a hugely successful and intensively studied variable selection technique, particularly for the ordinary least squares (OLS) problem (e.g. Efron et al. (2004); El Ghaoui et al. (2012); Friedman et al. (2010); Johnstone and Titterington (2009); Osborne et al. (2000a,b); Tibshirani (1996, 2013); Tibshirani et al. (2012); Zou and Hastie (2005)). Generalised linear models (GLMs) relax the implicit OLS assumption that the response variable is normally distributed and can be applied to, for instance, binomially distributed binary outcome data or poisson distributed count data (Nelder and Wedderburn, 1972). However, the most popular and efficient algorithm for $\ell_1$-penalised regression in GLMs uses a quadratic approximation to the log-likelihood function to map the problem back to an OLS problem and although it works well in practice, it is not guaranteed to converge to the optimal solution (Friedman et al., 2010). Here it is shown that calculating the maximum likelihood coefficient estimates for $\ell_1$-penalised regression in generalised linear models can be done via a coordinate descent algorithm consisting of successive soft-thresholding operations on the unpenalised maximum log-likelihood function without requiring an intermediate OLS approximation. Because this algorithm can be expressed entirely in terms of the natural formulation of the GLM, it is proposed to call it the natural coordinate descent algorithm.

To make these statements precise, let us start by introducing a response variable $Y \in \mathbb{R}$ and predictor vector $X \in \mathbb{R}^p$. It is assumed that $Y$ has a probability distribution from the exponential family, written in canonical form as

$$p(y \mid \eta, \phi) = h(y, \phi) \exp \left( \alpha(\phi) \{ y \eta - A(\eta) \} \right)$$

Research supported by Roslin Institute Strategic Grant funding from the BBSRC.
where \( \eta \in \mathbb{R} \) is the natural parameter of the distribution, \( \phi \) is a dispersion parameter and \( h, a > 0 \) and \( A \) convex are known functions. The expectation value of \( Y \) is a function of the natural parameter, \( E(Y) = A'(\eta) \), and linked to the predictor variables by the assumption of a linear relation \( \eta = X^T \beta \), where \( \beta \in \mathbb{R}^p \) is the vector of regression coefficients. It is tacitly assumed that \( \beta_1 \equiv 1 \) such that \( \beta_1 \) represents the intercept parameter. Suppose now that we have \( n \) observation pairs \( (x_i, y_i) \) (with \( x_{i1} = 1 \) fixed for all \( i \)). The minus log-likelihood of the observations for a given set of regression coefficients \( \beta \) under the GLM is given by

\[
H(\beta) = \frac{1}{n} \sum_{i=1}^{n} A(x_i^T \beta) - y_i(x_i^T \beta) = U(\beta) - w^T \beta
\]

where any terms not involving \( \beta \) have been omitted, \( U(\beta) = \frac{1}{n} \sum_{i=1}^{n} A(x_i^T \beta) \) is a convex function, \( w = \frac{1}{n} \sum_{i=1}^{n} y_i x_i \in \mathbb{R}^p \), and the dependence of \( U \) and \( w \) on the data \( (x_i, y_i) \) has been suppressed for notational simplicity. In the penalised regression setting, this cost function is augmented with \( \ell_1 \) and \( \ell_2 \) penalty terms to achieve regularity and sparsity of the minimum-energy solution, i.e. \( H \) is replaced by

\[
H(\beta) = U(\beta) - w^T \beta + \lambda \sum_{j=1}^{p} |\beta_j|^2 + \mu \sum_{j=1}^{p} |\beta_j|
\]

where \( \| \beta \|_2 = (\sum_{j=1}^{p} |\beta_j|^2)^{1/2} \) and \( \| \beta \|_1 = \sum_{j=1}^{p} |\beta_j| \) are the \( \ell_2 \) and \( \ell_1 \) norm, respectively, and \( \lambda \) and \( \mu \) are positive constants. The \( \ell_2 \) term merely adds a quadratic function to \( U \) which serves to make its Hessian matrix non-singular and it will not need to be treated explicitly in our analysis. Furthermore a slight generalisation is made where instead of a fixed parameter \( \mu \), a vector of predictor-specific penalty parameters \( \mu_j \) is used. This allows for instance to account for the usual situation where the intercept coefficient is unpenalised \( (\mu_1 = 0) \). The problem we are interested in is thus to find

\[
\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} H(\beta)
\]

with \( H \) a function of the form

\[
H(\beta) = U(\beta) - w^T \beta + \sum_{j=1}^{p} \mu_j |\beta_j|
\]

where \( U : \mathbb{R}^p \to \mathbb{R} \) is a smooth convex function, \( w \in \mathbb{R}^p \) is an arbitrary vector and \( \mu \in \mathbb{R}^p \), \( \mu \succ 0 \) is a vector of non-negative parameters. The notation \( u \succ v \) is used to indicate that \( u_j \geq v_j \) for all \( j \) and likewise the notation \( u \cdot v \) will be used to indicate elementwise multiplication, i.e. \( (u \cdot v)_j = u_j v_j \). The maximum of the unpenalised log-likelihood, considered as a function of \( w \), is of course the Legendre transform of the convex function \( U 
\]

\[
L(w) = \max_{\beta \in \mathbb{R}^p} \left\{ w^T \beta - U(\beta) \right\}
\]

and standard results from convex analysis (Boyd and Vandenberghe, 2004) imply that the unpenalised regression coefficients satisfy

\[
\hat{\beta}_0(w) = \arg\max_{\beta \in \mathbb{R}^p} \left\{ w^T \beta - U(\beta) \right\} = \nabla L(w),
\]

where \( \nabla \) is the usual gradient operator. We have the following key result:

**Theorem 1.** The solution \( \hat{\beta}(w, \mu) \) of

\[
\hat{\beta}(w, \mu) = \arg\min_{\beta \in \mathbb{R}^p} \left\{ U(\beta) - w^T \beta + \sum_{j=1}^{p} \mu_j |\beta_j| \right\}
\]
is given by
\[ \hat{\beta}(w, \mu) = \hat{\beta}_0(\hat{u}(w, \mu)) = \nabla L(\hat{u}(w, \mu)) \]
where \( \hat{u}(w, \mu) \) is the solution of the constrained convex optimisation problem
\[ \hat{u}(w, \mu) = \arg\min_{u \in \mathbb{R}^p : |u - w| \leq \mu} L(u). \tag{6} \]
Furthermore the sparsity patterns of \( \hat{\beta} \) and \( \hat{u} - w + \text{sgn}(\hat{\beta}) \cdot \mu \) are complementary,

\[ \hat{\beta}_j(w, \mu) \neq 0 \Leftrightarrow \hat{u}_j(w, \mu) = w_j - \text{sgn}(\hat{\beta}_j) \mu_j. \]

The proof of this Theorem consists of a straightforward application of Fenchel’s duality theorem and is provided in Appendix A.

2. Natural Coordinate Descent Algorithm

It is well-known that a cyclic coordinate descent algorithm for the \( \ell_1 \)-penalised optimisation problem in eq. (5) converges (Tseng, 2001). When only one variable is optimised at a time, keeping all others fixed, the equivalent variational problem in eq. (6) reduces to a remarkably simple soft-thresholding mechanism illustrated in Figure 1. More precisely, let \( U(\beta) \) be a smooth convex function of a single variable \( \beta \in \mathbb{R} \), \( w_0 = \arg\min_{\beta \in \mathbb{R}} L(u) \) and \( \sigma = \text{sgn}(w - w_0) \). The solution of the one-variable optimisation problem

\[ \hat{u} = \arg\min_{u \in \mathbb{R} : |u - w| \leq \mu} L(u), \]

with \( \mu \geq 0 \), can be expressed as follows. If \( |w - w_0| \leq \mu \) then \( \hat{u} = w_0 \) and hence

\[ \hat{\beta}(w, \mu) = 0. \]

Otherwise we must have

\[ \hat{u} = w - \sigma \mu \quad \text{and} \quad \hat{\beta}(w, \mu) = \hat{\beta}_0(w - \sigma \mu). \]

Hence the solution takes the form of a generalised ‘soft-thresholding’

\[ \hat{\beta}(w, \mu) = \begin{cases} \hat{\beta}_0(w - \sigma \mu) & |w - w_0| > \mu \\ 0 & |w - w_0| \leq \mu, \end{cases} \tag{7} \]

see also Figure 1. In other words, compared to the multivariate problem in Theorem A, where there remains ambiguity about the signs \( \text{sgn}(\hat{\beta}) \), in the one-variable case the sign is uniquely determined by the relative position of \( w \) and \( w_0 \).

Numerically solving the unpenalised one-variable problem is usually straightforward. First note that by assumption, \( U \) is differentiable and therefore it is itself the Legendre transform of \( L \). Hence

\[ w_0 = \arg\min_{u \in \mathbb{R}} L(u) = \arg\max_{\beta \in \mathbb{R}} \{ \beta u - L(u) \} \big|_{\beta = 0} = U'(0). \]

Likewise, and assuming there exists no analytic expression for \( L \), \( \hat{\beta}_0(w - \sigma \mu) = \arg\max_{\beta \in \mathbb{R}} \{ (w - \sigma \mu) \beta - U(\beta) \} \) can be found as the zero of the function

\[ f(\beta) = U'(\beta) - w + \sigma \mu. \]

For \( U \) convex, this is a monotonically increasing function of \( \beta \) and conventional one-dimensional root-finding algorithms converge quickly.

The \( p \)-dimensional natural coordinate descent algorithm simply consists of iteratively applying the above procedure to the one-dimensional functions

\[ U_j(\hat{\beta}_j) = U(\hat{\beta}_1, \ldots, \hat{\beta}_{j-1}, \hat{\beta}_j, \hat{\beta}_{j+1}, \ldots, \hat{\beta}_p) \]

where \( \hat{\beta} \in \mathbb{R}^p \) are the current coefficient estimates. Standard techniques can be used to make the algorithm more efficient by organising the calculations around the set of non-zero coefficients (Friedman et al., 2010), that is, after every complete cycle through all coordinates, the current set of non-zero coefficients is updated.
a. Unpenalised cost function

\[ U(\beta) - w\beta \]

b. Unpenalised Legendre, tangent

\[ L(v) = \max_{\beta \in \mathbb{R}} \{ v\beta - U(\beta) \} \]

c. Penalised cost functions, different \( \mu \)

\[ U(\beta) - w\beta + \mu_1|\beta| \]

d. Unpenalised Legendre, constrained minimiser

\[ L(v) = \max_{\beta \in \mathbb{R}} \{ v\beta - U(\beta) \} \]

Figure 1. Illustration of Theorem in one dimension. a. The unpenalised cost function \( U(\beta) - w\beta \) is a convex function of \( \beta \); the maximum-likelihood estimate \( \hat{\beta}_0 \) is its unique minimiser. b. The maximum-likelihood estimate is also equal to the slope of the tangent to the Legendre transform of \( U \) at \( w \). c. Every value of the \( \ell_1 \) penalty parameter \( \mu \) leads to a different cost function; for \( \mu = \mu_1 \) sufficiently small, the maximum-likelihood estimate \( \hat{\beta}_1 < \hat{\beta}_0 \) is non-zero while for sufficiently large \( \mu = \mu_2 \) it is exactly zero. d. The penalised problem can also be solved by minimising the unpenalised Legendre transform over the interval \([w - \mu, w + \mu]\); for \( w > w_0 \) and \( \mu_1 < w - w_0 \) the absolute minimiser of \( L \) is not included in this interval such that the constrained minimiser is the boundary value \( w - \mu_1 \) and the the maximum-likelihood estimate \( \hat{\beta}_1 \) equals the slope of the tangent at \( w - \mu_1 \), while for \( \mu_2 \geq w - w_0 > 0 \), the constrained minimiser is always the absolute minimiser which has a tangent with slope zero. Note that because \( L \) is convex, the slope at \( w - \mu_1 \) is always smaller than the slope at \( w \) (i.e. \( \hat{\beta}_1 < \hat{\beta}_0 \)). Similar reasoning applies when \( w < w_0 \).

until convergence before another complete cycle is run (see pseudocode in Appendix C).

An alternative method for updating \( \hat{\beta}_j \) is to use a quadratic approximation to \( U_j(\beta_j) \) around the current estimate of \( \hat{\beta}_j \), leading to a linear approximation for
NATURAL COORDINATE DESCENT FOR L1 REGRESSION

updating $\hat{\beta}_j$, i.e.

$$
\hat{\beta}_j^{(\text{new})} = \arg\max_{\beta \in \mathbb{R}} \{ \langle w_j - \sigma_j \mu_j \rangle \beta - U_j(\beta) \}
$$

$$
\approx \arg\max_{\beta \in \mathbb{R}} \{ \langle w_j - \sigma_j \mu_j \rangle \beta - U_j(\hat{\beta}_j) - \frac{1}{2} U''_j(\hat{\beta}_j) (\beta - \hat{\beta}_j)^2 \}
$$

$$
= \hat{\beta}_j + \frac{w_j - \sigma_j \mu_j - U'_j(\hat{\beta}_j)}{U''_j(\hat{\beta}_j)}.
$$

This approximation differs from the usual quadratic approximation \cite{friedman2010} by the fact that it still uses the \textit{exact} thresholding rule \cite{7}.

3. Numerical experiments

I implemented the natural coordinate descent algorithm for logistic regression in C with a Matlab interface (source code available from \url{http://glmnat.googlecode.com}). The penalised cost function for $\beta \in \mathbb{R}^p$ in this case is given by

$$
H(\beta) = U(\beta) - w^T \beta + \mu \sum_{j=2}^p |\beta_j|,
$$

where

$$
U(\beta) = \frac{1}{n} \sum_{i=1}^n \log \left(1 + e^{x_i^T \beta} \right), \quad w = \frac{1}{n} \sum_{i=1}^n y_i x_i^T,
$$

and $(x_i \in \mathbb{R}^p, y_i \in \{0, 1\}), i = 1, \ldots, n$ are the observations. Recall from Section 1 that $\beta_1$ is regarded as the (unpenalised) intercept parameter and therefore a fixed value of one ($x_{i1} = 1$) is added to every observation. As convergence criterion I used $\max_{j=1,\ldots,p} |\hat{\beta}_j^{(\text{new})} - \hat{\beta}_j^{(\text{old})}| < \epsilon$, where $\epsilon > 0$ is a fixed parameter. The difference is calculated at every iteration step when a single coefficient is updated and the maximum is taken over a full iteration after all, resp. all active, coefficients have been updated once.

To test the algorithm I used gene expression levels for 17,814 genes in 540 breast cancer samples (BRCA dataset) \cite{cancer2012b} and 20,531 genes in 266 colon and rectal cancer samples (COAD dataset) \cite{cancer2012a} as predictors for estrogen receptor status (BRCA) and early-late tumor stage (COAD), respectively (see Appendix B for details, processed data available from \url{http://glmnat.googlecode.com}). I compared the implementation of the natural coordinate descent algorithm against \texttt{glmnet} (version dated 30 Aug 2013) \cite{qian2013}, a Fortran-based implementation for Matlab of the coordinate descent algorithm for penalised regression in generalised linear models proposed by \cite{friedman2010}, which was found to be the most efficient in a comparison to various other softwares by the original authors \cite{friedman2010} as well as in an independent study \cite{yuan2010}. All analyses were run on a laptop with 2.7 GHz processor and 8 GB RAM using Matlab v8.2.0.701 (R2013b). Following \cite{friedman2010}, I considered a geometric path of regularisation parameters $\mu^{(k)} = \mu^{(1)}/m^{-k}$ where $\mu^{(1)} = \max_{j=2,\ldots,p} |w_j|$ is the smallest penalty that yields a solution where only the intercept parameter is non-zero, a value of $m = 100$ was used, and $k = 1, \ldots, m$, corresponding to the default choice in \texttt{glmnet}. To compare the output of two different algorithms over
the entire regularisation path, I considered the maximum relative score difference
\[
\max_{k=1,\ldots,m} \frac{H(\hat{\beta}^{(1,k)}) - H(\hat{\beta}^{(2,k)})}{H(\hat{\beta}^{(1,k)})}
\]
where \(\hat{\beta}^{(1,k)}\) and \(\hat{\beta}^{(2,k)}\) are the coefficient estimates obtained by the respective algorithms for the \(k\)th penalty parameter.

A critical issue when comparing algorithm runtimes are convergence threshold settings. Figure 2a shows the runtimes of the exact natural coordinate descent algorithm and its linear approximation (cf. Section 2 and Appendix C) and their maximum relative score difference for a range of values of the convergence threshold \(\epsilon\). The linear algorithm is about twice as fast as the exact algorithm and, as expected, both return numerically identical results within the accepted tolerance levels. For subsequent analyses only the linear algorithm was used. Because glmnet uses a different convergence criterion than the one used here, I ran the natural coordinate descent algorithm with a range of values for \(\epsilon\) and calculated the maximum relative score difference over the entire regularisation path with respect to the output of glmnet with default settings. Figure 2b shows that there is a dataset-dependent value for \(\epsilon\) where this difference is minimised and that the minimum difference is within the range observed when running glmnet with randomly permuted order of predictors. These minimising values \(\epsilon_{\text{BRCA}} = 6.3 \times 10^{-4}\) and \(\epsilon_{\text{COAD}} = 2.0 \times 10^{-3}\) were used for the subsequent comparisons. For these comparisons, I considered both ‘cold’ (each \(\hat{\beta}(\mu(k))\) calculated individually with initial value \(\hat{\beta} = 0\)) and ‘warm’ (each \(\hat{\beta}(\mu(k))\) calculated along the regularisation path \(\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(k)}\), each time using \(\hat{\beta}(\mu^{(l)})\) as initial values for the calculation of \(\hat{\beta}(\mu^{(l+1)})\)) starts. For glmnet, there is a clear advantage to using warm starts and, as also observed by Friedman et al. (2010), for smaller values of \(\mu\), it can be faster to compute along a regularisation path down to \(\mu\) than to compute the solution at \(\mu\) directly (Figure 2c,d). In contrast, the natural coordinate descent algorithm is much less sensitive to the use of warm starts (i.e. to the choice of initial value for \(\hat{\beta}\)) and it is considerably faster than glmnet when calculating solutions at single penalty parameter values (Figure 2c,d).

4. CONCLUSIONS

The popularity of \(\ell_1\)-penalised regression as a variable selection technique owes a great deal to the availability of highly efficient coordinate descent algorithms. For generalised linear models, the best existing algorithm uses a quadratic least squares approximation where the coordinate update step can be solved analytically as a linear soft-thresholding operation. This analytic solution has been understood primarily as a consequence of the quadratic nature of the problem. Here it has been shown however that in the dual picture where the penalised optimisation problem is expressed in terms of its Legendre transform, this soft-thresholding mechanism is generic and a direct consequence of the presence of an \(\ell_1\)-penalty term. Incorporating this analytic result in a standard coordinate descent algorithm leads to a method that is not only theoretically attractive and easy to implement, but also appears to offer practical advantages compared to the existing quadratic-approximation algorithm. In particular it is more robust to the choice of starting vector and therefore considerably more efficient when it is cold-started, i.e. when a solution is computed at set values of the \(\ell_1\)-penalty parameter as opposed to along a regularisation path of descending \(\ell_1\)-penalties. This can be exploited for instance
in situations where prior knowledge or other constraints dictate the choice of $\ell_1$-penalty parameter or in situations where distributing the computations for sweeping the $\ell_1$-penalty parameter space over multiple processors can lead to significant gains in computing time.
Appendix A. Proof of Theorem

With the notations introduced in Section 1, let \( F(\beta) = U(\beta) - w^T \beta \) and \( G(\beta) = \sum_{j=1}^{p} \mu_j |\beta_j| \). \( F \) and \( G \) are convex functions on \( \mathbb{R}^p \) which satisfy Fenchel’s duality theorem (Rockafellar, 1970)

\[
\min_{\beta \in \mathbb{R}^p} \{ F(\beta) + G(\beta) \} = \max_{u \in \mathbb{R}^p} \{ -F^*(u) - G^*(-u) \},
\]

where \( F^* \) and \( G^* \) are the Legendre transforms of \( F \) and \( G \) respectively. Standard calculations (Boyd and Vandenberghe, 2004) show that \( F^*(u) = L(u + w) \) and \( G^*(u) = 0 \) if \( |u| \leq \mu \) and \( G^*(u) = \infty \) otherwise. It follows that

\[
\min_{\beta \in \mathbb{R}^p} H(\beta) = - \min_{u \in \mathbb{R}^p: |u-w| \leq \mu} L(u) = -L(\hat{u}) = \min_{\beta \in \mathbb{R}^p} \{ U(\beta) - \hat{u}^T \beta \} \quad (9)
\]

where \( \hat{u} = \arg\min_{u \in \mathbb{R}^p: |u-w| \leq \mu} L(u) \). Denoting \( S = \{ u \in \mathbb{R}^p: |u-w| \leq \mu \} \), the minimiser \( \hat{u} \) must satisfy the optimality conditions (Boyd and Vandenberghe, 2004 §4.2.3): \( \hat{u} \in S \) and

\[
(v - \hat{u})^T \nabla L(\hat{u}) \geq 0 \quad \text{for all } v \in S. \quad (10)
\]

For any index \( j \), choose \( v_j \in S_j = \{ u \in \mathbb{R}: |w_j - u| \leq \mu_j \} \) arbitrary and set \( v_k = \hat{u}_k \) for \( k \neq j \). Then \( v \in S \) and by eq. (10),

\[
(v_j - \hat{u}_j) \frac{\partial L}{\partial u_j}(\hat{u}) \geq 0. \quad (11)
\]

Assume \( \frac{\partial L}{\partial u_j}(\hat{u}) \neq 0 \) and \( \hat{u}_j \neq w_j - \sigma_j \mu_j \), where \( \sigma_j = \sgn(\frac{\partial L}{\partial u_j}(\hat{u})) \). Then there exists \( \epsilon > 0 \) such that \( v_j = \hat{u}_j - \epsilon \sigma_j \in S_j \), but this contradicts eq. (11). Recall that if \( \hat{\beta}_0 = \arg\min_{\beta \in \mathbb{R}^p} \{ U(\beta) - \hat{u}^T \beta \} \), then \( \hat{\beta}_0 = \nabla L(\hat{u}) \). Hence we have shown that

\[
\hat{\beta}_{0,j} \neq 0 \Leftrightarrow \hat{u}_j = w_j - \sgn(\hat{\beta}_{0,j}) \mu_j. \quad (12)
\]

Denote \( I = \{ j: \hat{\beta}_{0,j} \neq 0 \} \). We find

\[
\hat{u}^T \hat{\beta}_0 = \sum_{j \in I} \hat{u}_j \hat{\beta}_{0,j} = \sum_{j \in I} [w_j - \sgn(\hat{\beta}_{0,j}) \mu_j] \hat{\beta}_{0,j} = w^T \hat{\beta}_0 - \sum_{j \in I} \hat{\beta}_{0,j} \mu_j
\]

and hence by eq. (9),

\[
\min_{\beta \in \mathbb{R}^p} H(\beta) = U(\hat{\beta}_0) - \hat{u}^T \hat{\beta}_0 = H(\hat{\beta}_0),
\]

i.e. \( \hat{\beta}_0 \) is also the unique minimiser of the penalised cost function \( H \). This concludes the proof of Theorem

\[\square\]

Appendix B. The Cancer Genome Atlas data processing details

B.1. Breast cancer data (BRCA). Processed data files were obtained from [https://tcga-data.nci.nih.gov/docs/publications/brca_2012/](https://tcga-data.nci.nih.gov/docs/publications/brca_2012/)

- Normalised expression data for 17,814 genes in 547 breast cancer samples (file BRCA.exp.547.med.txt).
- Clinical data for 850 breast cancer samples (file BRCA.Clinical.tar.gz).

540 samples common to both files had an estrogen receptor status reported as positive or negative in the clinical data. Estrogen receptor status was used as the binary response data \( Y \in \mathbb{R}^n, n = 540 \), and gene expression for all genes (+ one constant predictor) was used as predictor data \( X \in \mathbb{R}^{n \times p}, p = 17,815 \).
B.2. Colon and rectal cancer data (COAD). Processed data files were obtained from https://tcga-data.nci.nih.gov/docs/publications/coadread_2012/:

- Normalised expression data for 20,531 genes in 270 colon and rectal cancer samples (file crc_270_gene_rpkmdatac.txt).
- Clinical data for 276 colon and rectal cancer samples (file crc_clinical_sheet.txt).

266 samples common to both files had a tumor stage (from I to IV) reported in the clinical data. Early (I–II) and late (III–IV) stages were grouped and used as the binary response data $Y \in \mathbb{R}^n$, $n = 266$, and gene expression for all genes (+ one constant predictor) was used as predictor data $X \in \mathbb{R}^{n \times p}$, $p = 20,532$.

Algorithm 1 Main loop

Initialise $\hat{\beta} = 0$.

COMPLETECYCLE
while not converged do
    while not converged do
        ACTIVESETCYCLE
        end while
    COMPLETECYCLE
end while

Algorithm 2 Complete coordinate descent cycle

procedure COMPLETECYCLE
for $j = 1, \ldots, p$ do
    COORDINATEUPDATE($j$)
end for
end procedure

Algorithm 3 Active set coordinate descent cycle

procedure ACTIVESETCYCLE
for $j = 1, \ldots, p$ do
    if $\hat{\beta}_j \neq 0$ then
        COORDINATEUPDATE($j$)
    end if
end for
end procedure
Algorithm 4 Exact coordinate update

procedure COORDINATEUPDATE(j)
    Update $w_{0j} = U_j'(0)$.
    if $|w_j - w_{0j}| > \mu_j$ then
        $\hat{\beta}_j \leftarrow \text{zero of } U_j'(\cdot) - w_j + \text{sgn}(w_j - w_{0j})\mu_j$.
    else
        $\hat{\beta}_j \leftarrow 0$.
    end if
end procedure

Algorithm 5 Linear approximation for coordinate update

procedure COORDINATEUPDATELINEAR(j)
    Update $w_{0j} = U_j'(0)$.
    if $|w_j - w_{0j}| > \mu_j$ then
        $\hat{\beta}_j \leftarrow \hat{\beta}_j + \frac{w_j - \text{sgn}(w_j - w_{0j})\mu_j}{U_j'(\hat{\beta}_j)}U_j''(\hat{\beta}_j)$
    else
        $\hat{\beta}_j \leftarrow 0$.
    end if
end procedure

References

Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics* 32(2), 407–499.

El Ghaoui, L., V. Viallon, and T. Rabbani (2012). Safe feature elimination for the lasso and sparse supervised learning problems. *Pacific Journal of Optimization* 8, 667–698.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33(1), 1.

Johnstone, I. and D. Titterington (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367(1906), 4237–4253.

Nelder, J. and R. Wedderburn (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 370–384.

Osborne, M., B. Presnell, and B. Turlach (2000a). On the lasso and its dual. *Journal of Computational and Graphical Statistics* 9(2), 319–337.

Osborne, M. R., B. Presnell, and B. A. Turlach (2000b). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis* 20(3), 389–403.

Qian, J., T. Hastie, J. Friedman, R. Tibshirani, and N. Simon (2013). Glmnet for Matlab. [http://www.stanford.edu/~hastie/glmnet_matlab/](http://www.stanford.edu/~hastie/glmnet_matlab/)

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

The Cancer Genome Atlas Network (2012a). Comprehensive molecular characterization of human colon and rectal cancer. *Nature* 487, 330–337.

The Cancer Genome Atlas Network (2012b). Comprehensive molecular portraits of human breast tumours. *Nature* 490, 61–70.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Tibshirani, R. (2013). The lasso problem and uniqueness. *Electronic Journal of Statistics* 7, 1456–1490.

Tibshirani, R., J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. Tibshirani (2012). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74(2), 245–266.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications* 109(3), 475–494.

Yuan, G.-X., K.-W. Chang, C.-J. Hsieh, and C.-J. Lin (2010). A comparison of optimization methods and software for large-scale L1-regularized linear classification. *Journal of Machine Learning Research* 9999, 3183–3234.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2), 301–320.

---

**E-mail address:** tom.michoel@roslin.ed.ac.uk