Biometrika (2009), **96**, 2, *pp.* 323–337
doi: 10.1093/biomet/asp013

Printed in Great Britain

A generalized Dantzig selector with shrinkage tuning

BY GARETH M. JAMES AND PETER RADCHENKO

Marshall School of Business, University of Southern California,
Los Angeles, California 90089, U.S.A.
gareth@usc.edu radchenk@marshall.usc.edu

SUMMARY

The Dantzig selector performs variable selection and model fitting in linear regression. It uses an L_1 penalty to shrink the regression coefficients towards zero, in a similar fashion to the lasso. While both the lasso and Dantzig selector potentially do a good job of selecting the correct variables, they tend to overshrink the final coefficients. This results in an unfortunate trade-off. One can either select a high shrinkage tuning parameter that produces an accurate model but poor coefficient estimates or a low shrinkage parameter that produces more accurate coefficients but includes many irrelevant variables. We extend the Dantzig selector to fit generalized linear models while eliminating overshrinkage of the coefficient estimates, and develop a computationally efficient algorithm, similar in nature to least angle regression, to compute the entire path of coefficient estimates. A simulation study illustrates the advantages of our approach relative to others. We apply the methodology to two datasets.

Some key words: Dantzig selector; DASSO; Double Dantzig; Generalized linear model; Interpolated Dantzig; Lasso; Ridge Dantzig; Variable selection.

1. INTRODUCTION

In the traditional linear regression model,

$$Y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i \quad (i = 1, \ldots n), \tag{1}$$

a common problem is that of variable selection, i.e. determining which β_j are nonzero. Recent methods include the lasso (Tibshirani, 1996; Chen et al., 1998), the adaptive lasso (Zou, 2006), smoothly clipped absolute deviation (Fan & Li, 2001), the elastic net (Zou & Hastie, 2005), the relaxed lasso (Meinshausen, 2007), the Dantzig selector (Candès & Tao, 2007) and variable inclusion and shrinkage algorithms (Radchenko & James, 2008). Many of these methods use an L_1 penalty on the regression coefficients, which has the effect of shrinking some coefficients towards zero and setting others to exactly zero, thereby performing variable selection.

This paper concerns a less well studied problem, that of variable selection methods for generalized linear models (McCullagh & Nelder, 1989), of which logistic regression is an example. There has been previous work in this area. Tibshirani (1996) briefly discusses using the lasso to fit generalized linear models. Others have used L_1 penalties to fit logistic regression models; see Shevade & Keerthi (2003), Genkin et al. (2006), and an unpublished technical report from the University of Adelaide by J. Lokhorst. In addition, Zhao & Yu (2007) and Park & Hastie (2007) develop methods for fitting the entire coefficient path for generalized linear and other models with L_1 penalties.

We use the Dantzig selector, which works by minimizing the L_1 norm of β subject to an L_∞ constraint on the correlation of the residuals with the predictors. We make three main contributions. First, the Dantzig selector was not designed for generalized linear models, so we develop an extension, the generalized Dantzig selector, which works by reformulating the Dantzig selector criteria in terms of the maximum likelihood equations, thereby suggesting a natural extension to arbitrary response distributions. Second, we present a path algorithm for simultaneously computing the generalized Dantzig selector coefficients for all values of the tuning parameter, λ, in an efficient manner. This makes it practical to perform crossvalidation on λ even for very large datasets.

The third contribution is the introduction of three closely related new methods, the double Dantzig, ridge Dantzig and interpolated Dantzig methods, designed for fitting generalized linear models with large numbers of predictors but sparse coefficient vectors. The Dantzig selector and lasso tend either to select the correct variables and produce poor coefficient estimates or to generate more accurate coefficients at the expense of including many irrelevant variables. Our methods eliminate this drawback by using two-step procedures. For all three, the first step involves using the generalized Dantzig selector to select a candidate set of predictors. In the second step, we estimate the coefficients by fitting various models to the candidate predictors: the double Dantzig reapplies the generalized Dantzig selector, with a new tuning parameter; the ridge Dantzig uses a generalized linear model version of ridge regression; the interpolated Dantzig uses a weighted average of the original generalized Dantzig selector coefficients and the coefficients from a generalized linear model fit to the candidate predictors.

2. METHODOLOGY

2·1. *The Dantzig selector*

The Dantzig selector (Candès & Tao, 2007) was designed for linear regression models such as (1) with large p but a sparse set of coefficients, in that most of the β_js are zero. Assume that in the linear regression model (1) the columns of the design matrix, X, have been normalized. The Dantzig selector estimator, $\hat{\beta}$, is the solution to

$$\min_{\tilde{\beta}\in\mathcal{B}} ||\tilde{\beta}||_1 \quad \text{subject to} \quad \left|X_j^{\mathrm{T}}(Y - X\tilde{\beta})\right| \leqslant \lambda \quad (j = 1, \ldots, p), \tag{2}$$

where $||\cdot||_1$ represents the L_1 norm, X_j is the jth column of the design matrix, λ is a tuning parameter and $\mathcal{B}$ represents the set of possible values for β, usually taken to be a subset of $\mathbb{R}^p$. The L_1 minimization produces coefficient estimators that are exactly zero in a similar fashion to the lasso and hence can be used as a variable selection tool. Candès & Tao (2007) provide several examples of real-world problems involving large values of p, where the Dantzig selector performs well.

In this set-up X_j is assumed to have norm one, which is rarely the case in practice. However, this difficulty is easily resolved by reparameterizing (1) such that the X_js have norm one. Let $d_j = ||X_j||_2$. Then computing $\hat{\beta}$ for the standardized X_js and returning $\hat{\beta}_j/d_j$ produces the Dantzig selector estimator for β_j on the original scale.

For Gaussian error terms, (2) can be rewritten as

$$\min_{\tilde{\beta}\in\mathcal{B}} ||\tilde{\beta}||_1 \quad \text{subject to} \quad |l_j'(\tilde{\beta})| \leqslant \lambda/\sigma \quad (j = 1, \ldots, p), \tag{3}$$

where l_j' is the partial derivative of the loglikelihood function with respect to β_j and $\sigma^2 = \mathrm{var}(\varepsilon_i)$. Hence, for $\lambda = 0$, (3) will return the maximum likelihood estimator. Alternatively, the constraint in (2) is equivalent to $X^{\mathrm{T}}Y = X^{\mathrm{T}}X\hat{\beta}$, which is simply the least squares normal equation. For $\lambda > 0$ the Dantzig selector searches for the $\tilde{\beta}$ with the smallest L_1 norm that, within a fixed tolerance, satisfies the normal equations characterizing the maximum likelihood solution, i.e. the sparsest $\tilde{\beta}$ that is still reasonably consistent with the observed data. Even for $p > n$, in which case the likelihood equation has infinitely many solutions, this approach can still hope to identify the correct solution, provided β is sparse, because it attempts only to locate the sparsest $\tilde{\beta}$ close to the peak of the likelihood function. One might imagine that minimizing the L_0 norm, which counts the number of nonzero components of a vector, would be more appropriate than the L_1 norm. However, directly minimizing the L_0 norm is computationally intractable and one can show that, under suitable conditions, the L_1 norm will also provide a sparse solution.

The Dantzig selector has some nice properties. The first is that (2) can be formulated as a linear programming problem, so standard software can easily be used. For example, in § 5 we cope with a dataset with over 2000 observations and almost 1500 variables. Recently, James et al. (2009) developed the Dantzig selector with sequential optimization, or DASSO, algorithm for fitting an entire coefficient path of the Dantzig selector. The algorithm efficiently constructs a piecewise linear path through a sequential simplex-like algorithm that identifies the path breakpoints and solves the corresponding linear programmes. It is more efficient than standard linear programming when the true number of predictors is sparse, though interior-point methods may be preferable when there are many influential predictors. The second useful property is theoretical. Candès & Tao (2007) prove a tight nonasymptotic bound on the error in the estimator for β that is within a factor of $\log p$ of the error rate achieved if the true predictors are assumed known. Since $\log p$ grows very slowly, the Dantzig selector only pays a small price for adaptively choosing the significant variables.

2·2. *Generalized Dantzig selector*

In our generalized linear model framework for a random variable Y, with distribution

$$p(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\},$$

we model the relationship between predictor and response as $g(\mu_i) = \sum_{j=1}^{p} X_{ij}\beta_j$, where $\mu_i = E(Y;\theta_i,\phi) = b'(\theta_i)$ and g is referred to as the link function. Common examples of g include the identity link used for normal response data and the logistic link used for binary response data. The coefficient vector β is generally estimated using maximum likelihood. For notational simplicity we will assume that g is chosen as the canonical link, though all the ideas generalize naturally to other link functions. With the canonical link function, the score statistics are given by

$$l'_j(\hat{\beta}) = X_j^{\mathrm{T}}(Y - \hat{\mu}) \quad (j = 1,\ldots,p), \tag{4}$$

where $\hat{\mu} = g^{-1}(X\hat{\beta})$. Hence the maximum likelihood estimator satisfies $X_j^{\mathrm{T}}(Y - \hat{\mu}) = 0$ $(j = 1,\ldots,p)$, which can be solved by an iteratively weighted least squares algorithm. However, when p is large relative to n, the maximum likelihood approach becomes undesirable. First, solving (4) will not produce any zero coefficients, so no variable selection is performed. As a result, the final model is less interpretable and probably less accurate. Second, for large p the variance of the estimated coefficients will become large and when $p > n$, equation (4) has no unique solution.

We address both these limitations by extending the Dantzig selector to the generalized linear models setting. In § 2·1 we observed that the constraints in the standard Dantzig selector can be written in terms of a Gaussian likelihood function. Hence a natural generalization involves formulating the optimization criterion in terms of a generalized linear models likelihood. In

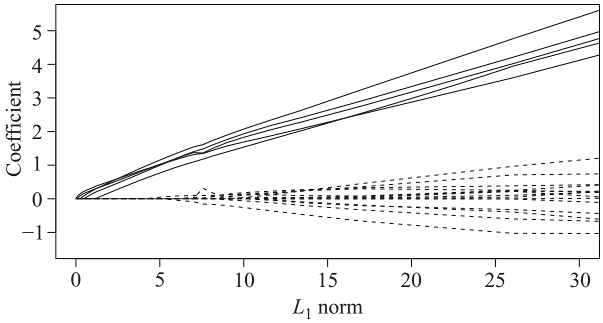

Fig. 1. Example 1. Plot of estimated coefficients for different values of λ. The solid lines represent the variables to which the response is related. The dashed lines correspond to the remaining predictors.

particular, by combining (3) and (4) we produce our generalized Dantzig selector criterion,

$$\min_{\tilde{\beta}\in\mathcal{B}} \|\tilde{\beta}\|_1 \quad \text{subject to} \quad \left|X_j^{\mathrm{T}}(Y-\tilde{\mu})\right| \leqslant \lambda \quad (j=1,\ldots,p), \tag{5}$$

where $\tilde{\mu} = g^{-1}(X\tilde{\beta})$. For the Gaussian distribution with the identity link function, the procedure reduces to the standard Dantzig selector. For binary {0, 1} responses, we can solve (5) using a logistic link function. The generalized Dantzig selector enjoys all the important properties of the Dantzig selector. In particular, it works well even when $p > n$ and produces sparse coefficient estimates, so it can be used for variable selection. In addition, in § 3 we show that the computational advantages of the Dantzig selector are preserved, and the entire coefficient path can be computed efficiently.

Example 1. Figure 1 provides an example of the generalized Dantzig selector coefficient estimates from a simulated dataset with Bernoulli responses and a logistic link function. The data were generated as $n = 100$ observations with $p = 20$ predictors with a design matrix produced using independent and identically distributed Gaussian random variables. All but the first five components of the true coefficient vector were zero; that is, the last 15 variables were independent of the response. This was a fairly hard problem because there were many variables, relatively few observations, and each response provided relatively little information because it only took two possible values. As noted previously, $\lambda = 0$ gives the standard generalized linear model fit. However, as λ grows, the generalized Dantzig selector constraint relaxes until the point where all the coefficients are estimated as zero. Figure 1 plots the estimated coefficients for various values of λ with the L_1 norm of the coefficients on the x-axis. Each line represents the coefficient for a single predictor variable with the thicker solid lines relating to the five true predictors and dashed lines for the remaining 15 unimportant variables. Despite the relatively small amount of information provided by the responses, the generalized Dantzig selector clearly identifies the correct five variables first with a significant break before adding the remaining variables. However, in the region where the correct variables are chosen, the estimated coefficients are approximately one, far below the true values of five. This example illustrates that, as with the lasso and the Dantzig selector, the generalized Dantzig selector has a tendency to overshrink the coefficients. We address this limitation in the following section.

2·3. *Shrinkage tuning methods*

As we illustrate in the simulation study of § 4, for models with sparse sets of nonzero coefficients, one can either select a high-shrinkage tuning parameter that produces an accurate model but poor coefficient estimates or a low-shrinkage tuning parameter that produces more accurate coefficients but includes many irrelevant variables. The latter case is the norm when crossvalidation is used to select λ.

Here we propose three methods that eliminate this trade-off by implementing the following two-stage fitting method.

Step 1. Select two tuning parameters, λ and ϕ.

Step 2. Fit the generalized Dantzig selector to the response, Y, using predictors, $X_1, \ldots, X_p$, and tuning parameter, λ.

Step 3. Identify the $q \leqslant p$ variables with nonzero coefficients, $X_1^*, \ldots, X_q^*$.

Step 4. Use the predictors $X_1^*, \ldots, X_q^*$ to estimate the corresponding coefficients.

In this paper we investigate three methods for implementing the final step.

Method 1. Reapply the generalized Dantzig selector to the response Y, using predictors $X_1^*, \ldots, X_q^*$ and tuning parameter, ϕ. We call this the double Dantzig method.

Method 2. Fit a generalized linear model version of ridge regression to the response Y, using predictors, $X_1^*, \ldots, X_q^*$, and tuning parameter, ϕ. This is implemented using an iterated reweighted algorithm, but instead of applying least squares to the adjusted dependent variable, the coefficients are estimated using ridge regression. We call this the ridge Dantzig method.

Method 3. Linearly interpolate between the coefficient estimates obtained using the generalized Dantzig selector in Step 2 above and the coefficient estimates from generalized linear models applied to the predictors, $X_1^*, \ldots, X_q^*$. The tuning parameter ϕ determines the relative weighting of the two sets of coefficients. We call this the interpolated Dantzig method.

The interpolated Dantzig method shares some similarities with the relaxed lasso (Meinshausen, 2007) since both methods estimate the coefficients using an interpolation between an initial, highly shrunk fit, and a final, unshrunk fit. The key differences are that the relaxed lasso uses the lasso and it has only been applied to standard linear regression models rather than the generalized linear models framework. In this sense the interpolated Dantzig method can be regarded as an extension of the relaxed lasso.

Generally, λ will be set to generate a high degree of regularization, while ϕ will be chosen to ensure relatively little shrinkage of the final coefficients. The key idea behind all three methods is that, by using two different levels of shrinkage, one high and the other low, we can both select an accurate model and produce accurate coefficient estimates; see §§ 4 and 5 for illustrations.

A key issue is the choice of λ and ϕ. In § 3, we introduce a path algorithm for computing the generalized Dantzig selector for each value of λ. Hence, for any given pair of λ and ϕ, we can efficiently compute the crossvalidated deviance. In this paper, we compute the crossvalidated deviance on a grid of values for the two tuning parameters and select the pair corresponding to the lowest deviance. When prediction accuracy is the final goal, crossvalidation is natural. When inference on the regression coefficients is the goal, this is less clear. However, crossvalidation is quite standard in such settings, and our simulation results from § 4 suggest that such a method can work well even when estimating the regression coefficients.

3. Fitting the models

3·1. *Computing the generalized Dantzig selector*

Unless g is the identity function, the constraints in (5) are nonlinear, so linear programming software cannot be used directly to compute the generalized Dantzig selector solution. In the usual generalized linear model setting, an iterative weighted least squares algorithm is used to solve (4). In particular, given the current estimate for $\hat{\beta}$, an adjusted dependent variable $Z_i = \sum_{j=1}^{p} X_{ij}\hat{\beta}_j + (Y_i - \hat{\mu}_i)/V_i$ is computed, where $V_i = b''(\theta_i)$ is the variance of Y_i. A new estimate for β is then computed using weighted least squares, i.e. $X_j^{\mathrm{T}} W(Z - X\hat{\beta}) = 0$, for $j = 1, \ldots, p$, where $W = \mathrm{diag}(V_1, \ldots, V_n)$. This step is iterated until $\hat{\beta}$ converges.

An analogous iterative approach works well when computing the generalized Dantzig selector solution.

Step 1. At the $(k+1)$th iteration, let $V_i = b''(\hat{\theta}_i^{(k)})$ and $Z_i = \sum_{j=1}^{p} X_{ij}\hat{\beta}_j^{(k)} + (Y_i - \hat{\mu}_i^{(k)})/V_i$, where (k) denotes the corresponding estimate from the kth iteration.

Step 2. Let $Z_i^* = Z_i \surd V_i$ and $X_{ij}^* = X_{ij} \surd V_i$.

Step 3. Use the Dantzig selector to compute $\hat{\beta}^{(k+1)}$ with Z^* as the response and X^* as the design matrix.

Step 4. Repeat Steps 1–3 until convergence.

Step 3 is equivalent to

$$\min_{\tilde{\beta} \in \mathcal{B}} \|\tilde{\beta}\|_1 \quad \text{subject to} \quad \left|X_j^{\mathrm{T}} W(Z - X\tilde{\beta})\right| \leqslant \lambda \quad (j = 1, \ldots, p), \tag{6}$$

which can be solved by linear programming. In addition, once the algorithm has converged, $\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)}$ so $|X_j^{\mathrm{T}} W(Z - X\hat{\beta}^{(k+1)})| = |X_j^{\mathrm{T}} W W^{-1}(Y - \hat{\mu})| = |X_j^{\mathrm{T}}(Y - \hat{\mu})|$ and hence the solution to (6) will also be in the feasible region for (5).

As with the standard generalized linear models iterative algorithm, there is no theoretical guarantee of convergence. However, in practice we have found that, for sparse solutions, our algorithm generally converges very rapidly and the computation time is only a small multiple of that for the standard Dantzig selector. For large p and low values of λ, near the generalized linear models solution, the algorithm does not always converge, though in this circumstance it usually iterates between two similar estimates so one can choose either. We do not see this as a significant limitation because we are most interested in producing relatively sparse models, for which the algorithm always performs well.

3·2. *The generalized Dantzig selector path algorithm*

Efron et al. (2004) demonstrated that the coefficient path of the lasso, i.e. the set of all possible coefficient vectors for different values of the tuning parameter λ was piecewise linear. They introduced a highly efficient algorithm called least angle regression and showed that it could fit the entire lasso path. More recently James et al. (2009) developed the DASSO algorithm, a generalization of least angle regression, which can fit the coefficient paths for both the lasso and the Dantzig selector. Park & Hastie (2007) also use a variant of least angle regression to produce an approximation of the path from a generalized linear models version of lasso. One key advantage of all these approaches is that they allow one to perform crossvalidation on a fine grid of tuning parameters in a computationally efficient manner. Here we introduce our own algorithm for fitting the generalized Dantzig selector coefficient path. Our algorithm is a generalization of

DASSO and computes the generalized Dantzig selector path in a similar fashion to that of Park & Hastie's algorithm. As with the latter, our algorithm provides an approximation to the true path.

By analogy with DASSO, least angle regression and Park & Hastie's method, the generalized Dantzig selector algorithm progresses in piecewise linear steps. At the start of each step, the variables that have maximum covariance with the residual vector are identified. These variables make up the current 'active set' $\mathcal{A}$ and determine the optimal direction in which the coefficient vector β should move. We then update $\mathcal{A}$ and compute an approximate value of the distance that can be travelled before the active set changes. At this point we 'correct' the coefficient vector by solving the generalized Dantzig selector optimization problem for given sets of active constraints and nonzero coefficients. An appealing feature of such a correction is that an optimization package is not required. We outline the main steps of our algorithm below and provide the details of Steps 3–5 in the Appendix.

Step 1. Initialize $\beta_j^1 = 0$, for $j = 1, \ldots, p$. Set $\mathcal{A}^1 = \emptyset$ and $l = 1$.

Step 2. Let $c = X^{\mathrm{T}}(Y - \mu^l)$ and $C^l = \arg\max_j |c_j|$, where c_j is the jth component of c. Augment $\mathcal{A}^l$ with any variables not currently in $\mathcal{A}^l$ for which $|c_j| = C^l$.

Step 3. *Direction step.* Based on the current active set $\mathcal{A}^l$ and coefficient estimates β^l, compute the p-dimensional direction vector h^l that minimizes the rate of increase in $\|\beta^l\|_1$ relative to the decrease in $\|X^{\mathrm{T}}(Y - \mu^l)\|_\infty$.

Step 4. *Distance step.* Form $\mathcal{A}^{l+1}$ by updating $\mathcal{A}^l$. Compute γ^l, an approximate distance to travel in the direction h^l until a new variable enters the active set or one coefficient hits zero. Set $\tilde{\beta}^{l+1} = \beta^l + \gamma^l h^l$.

Step 5. *Correction step.* Produce the corrected coefficient vector β^{l+1} based on $\tilde{\beta}^{l+1}$.

Step 6. Set $l \leftarrow l + 1$. Repeat Steps 2–5 until $C^l = 0$.

The entire coefficient path can then be constructed by linearly interpolating $\beta^1, \ldots, \beta^L$, where L denotes the number of steps taken by the algorithm. Steps 3 and 4 above closely mirror the corresponding direction and distance steps in DASSO and essentially provide a linear approximation to the generalized Dantzig selector path. Step 5 is a corrector step to adjust for the lack of linearity.

Figure 2 shows the first segment of the generalized Dantzig selector path for the spam dataset, see Hastie et al. (2001), produced by both our path algorithm and the generalized Dantzig selector on a fine grid. Although the generalized Dantzig selector coefficient path may not be exactly piecewise linear, the two paths are practically identical.

4. Simulation study

In this section we present the results from several simulation studies that we conducted to compare the double Dantzig, ridge Dantzig and interpolated Dantzig methods with the generalized Dantzig selector and four other possible approaches: standard generalized linear models, the oracle, the Gauss Dantzig and Park & Hastie's method. In the oracle method, which represents a gold standard that cannot be used in practice, it is assumed that the important predictors are known and the generalized linear models procedure is used to estimate their coefficients. The Gauss Dantzig approach is suggested by Candès & Tao (2007), who first used the Dantzig selector as a variable selection tool to identify the important predictors and then used standard linear regression, on the reduced set of variables, to produce the final coefficient estimates. In

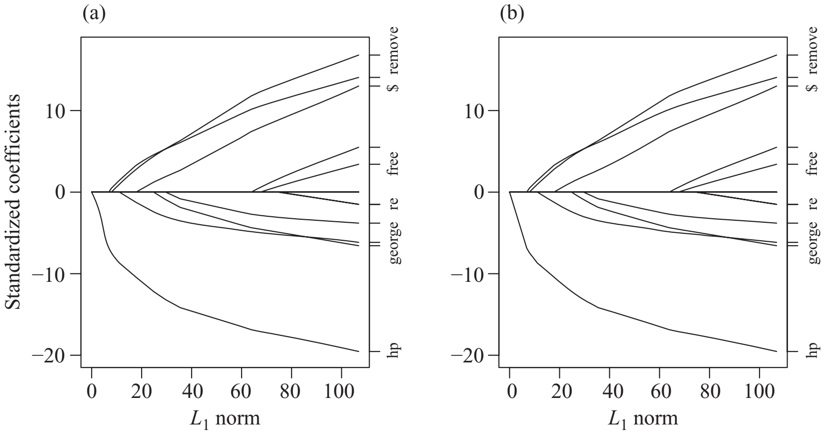

Fig. 2. Spam data. Plots of generalized Dantzig selector coefficient paths produced by (a) the path algorithm and (b) the generalized Dantzig selector on a grid.

our setting, this process involves using the generalized Dantzig selector to select the variables and then fitting a standard generalized linear model method on the subset. The Gauss Dantzig method is a special case of the double Dantzig method with ϕ set to zero. Finally, Park & Hastie's method uses the R package glmpath to compute the coefficients that minimize the generalized linear models loglikelihood subject to an L_1 penalty on β, i.e. a generalized linear models version of the lasso (Park & Hastie, 2007).

We tested these eight approaches in nine sets of simulations. All simulated datasets contained $n = 100$ observations and either $p = 20$, $p = 50$ or $p = 105$ predictors. In all cases only the first five predictors had a relationship with the response. For each of the nine sets of simulations, we generated 100 training datasets and 100 corresponding validation datasets. The methods were fitted using the training data, while the various tuning parameters were chosen so as to minimize the deviance on the validation datasets. For the double Dantzig method λ and ϕ were chosen as the values that jointly minimized the deviance on the validation data using a two-dimensional grid of possible values. The tuning parameters for the ridge Dantzig and interpolated Dantzig methods were separately chosen in an identical fashion. Similarly, the single tuning parameters for the Gauss and Park & Hastie's methods were the values minimizing the validation deviance. Finally, we used Park & Hastie's tuning parameter for the generalized Dantzig selector to maintain comparability between the two methods.

The first three sets of simulations used a binary response distribution with $p = 20$, $p = 50$ or $p = 105$ predictors. The design matrices X were generated using independent and identically distributed standard normal random variables and the first five regression coefficients were all set to the value three. The second three sets of simulations were identical to the first except that the design matrix was generated from normal random variables with a correlation coefficient of 0·2 between each pair of predictors. The final three sets of simulations used a Poisson response distribution, independent and identically distributed standard normal random variables for the design matrices and regression coefficients set to one rather than three.

Figure 3 provides a graphical representation of the root mean squared errors between the estimated and true coefficients over the 100 simulation runs for each simulation set. We have excluded the boxplots for the generalized linear models fits because the errors were so large. In Figs. 3(a), (b) and (c), for the Binomial simulations with independent and identically distributed entries in the design matrix, all three of our methods performed very well. In some cases they even

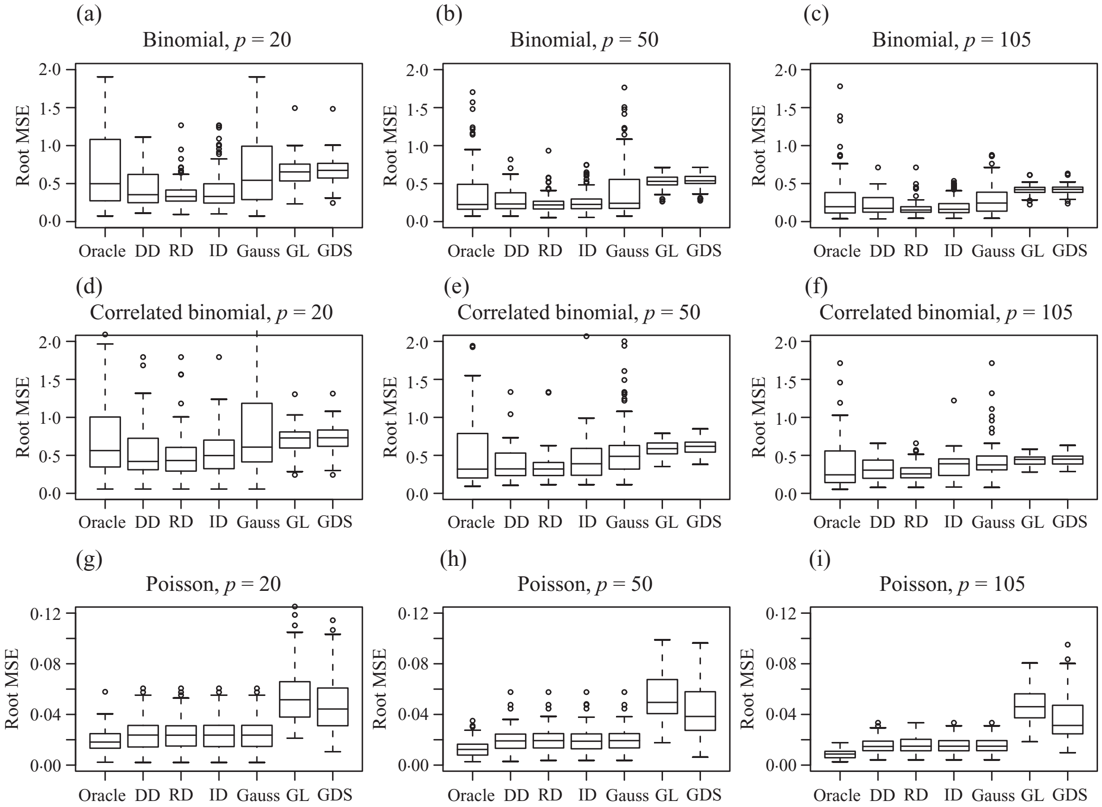

Fig. 3. Simulation study. Boxplots of the root mean squared errors between the estimated and true coefficients for the seven different methods over the 100 simulation runs for each set of simulations. Oracle, oracle method; DD, double Dantzig method; RD, ridge Dantzig method; ID, interpolated Dantzig method; Gauss, Gauss Dantzig method; GL, Park & Hastie's method; GDS, generalized Dantzig selector.

outperformed the oracle method because, while the oracle approach fitted the correct variables, it did not shrink the regression coefficients. The Gauss method performed similarly to the oracle but was inferior to our three methods. Finally, Park & Hastie's method and the generalized Dantzig selector provided almost identical results to each other but, with the exception of generalized linear models, were worse than the other methods. This similarity is consistent with the theory and empirical results in Efron et al. (2007), Meinshausen et al. (2008), James et al. (2009) and others, which show strong connections between the lasso and Dantzig selector. In terms of the median error rate, the double Dantzig, ridge Dantzig and interpolated Dantzig methods were almost identical. However, the second appeared to have slightly more stable results and hence less of a tail in terms of error rates.

Figures 3(d), (e) and (f) illustrate the Binomial simulations with correlated entries in the design matrix. The addition of correlations among the predictors made the problem more difficult and increased the errors for all seven methods. However, in general the relative performances among the methods were very similar to the uncorrelated case. Perhaps the worst affected was the interpolated Dantzig approach, which became more unstable and experienced a relatively larger increase in its typical error. Figures 3(g), (h) and (i) provide the Poisson results. Interestingly, the double Dantzig, ridge Dantzig, interpolated Dantzig and Gauss methods exhibited almost identical performances. The oracle approach provided slightly better results but, of course, could not be used in practice. The performances of Park & Hastie's method and the generalized Dantzig

Table 1. *Median root mean squared error, false positive rate $F+$ and false negative rate $F-$, for eight different methods and three different response distributions, using* 100 *simulations each with $p = 20$, $p = 50$ and $p = 105$. The sample size for each simulation was $n = 100$*

		$p = 20$			$p = 50$			$p = 105$		
Response	Method	RMSE	$F+$	$F-$	RMSE	$F+$	$F-$	RMSE	$F+$	$F-$
Binomial	Oracle	0·499	0·000	0·000	0·225	0·000	0·000	0·196	0·000	0·000
	DD	0·354	0·047	0·000	0·232	0·013	0·000	0·175	0·010	0·004
	RD	0·329	0·046	0·002	0·219	0·011	0·002	0·152	0·009	0·006
	ID	0·331	0·050	0·000	0·226	0·013	0·002	0·162	0·010	0·004
	Gauss	0·544	0·026	0·022	0·241	0·009	0·020	0·244	0·006	0·030
	GLasso	0·654	0·585	0·008	0·529	0·354	0·000	0·420	0·209	0·010
	GDS	0·674	0·579	0·006	0·537	0·365	0·000	0·424	0·209	0·006
	GLM	46·224	1·000	0·000	8·452	1·000	0·000			
Correlated binomial	Oracle	0·562	0·000	0·000	0·318	0·000	0·000	0·244	0·000	0·000
	DD	0·420	0·071	0·000	0·322	0·052	0·006	0·305	0·035	0·018
	RD	0·432	0·074	0·000	0·321	0·048	0·008	0·256	0·033	0·026
	ID	0·497	0·079	0·000	0·389	0·056	0·008	0·389	0·039	0·026
	Gauss	0·608	0·058	0·012	0·487	0·036	0·066	0·375	0·019	0·110
	GLasso	0·729	0·515	0·000	0·588	0·298	0·000	0·447	0·182	0·010
	GDS	0·731	0·525	0·000	0·618	0·298	0·000	0·449	0·180	0·016
	GLM	29·485	1·000	0·000	7·606	1·000	0·000			
Poisson	Oracle	0·018	0·000	0·000	0·012	0·000	0·000	0·009	0·000	0·000
	DD	0·024	0·082	0·000	0·019	0·067	0·000	0·015	0·045	0·000
	RD	0·024	0·082	0·000	0·019	0·066	0·000	0·015	0·045	0·000
	ID	0·024	0·084	0·000	0·019	0·069	0·000	0·015	0·046	0·000
	Gauss	0·024	0·081	0·000	0·019	0·068	0·000	0·015	0·045	0·000
	GLasso	0·052	0·601	0·014	0·049	0·340	0·052	0·046	0·222	0·048
	GDS	0·044	0·497	0·016	0·038	0·266	0·062	0·031	0·164	0·046
	GLM	0·068	1·000	0·000	0·143	1·000	0·000			

DD, double Dantzig; RD, ridge Dantzig; ID, interpolated Dantzig; Gauss, Gauss method; GLasso, Park & Hastie's method; GDS, generalized Dantzig selector; GLM, generalized linear models; RMSE, root mean squared error.

selector were both markedly inferior to the other five methods. However, while they acted in almost identical fashions for the Binomial cases, the generalized Dantzig selector performed slightly better than Park & Hastie's method for the Poisson response. This is interesting because previous empirical comparisons of the lasso and Dantzig selector have failed to find a scenario where the latter outperformed the former.

Numerical results for the nine simulations are given in Table 1. For each combination of p and response distribution, we give the median root mean squared error, as in Fig. 3. We also provide the false positive rate, i.e. the proportion of unrelated variables that are estimated to have nonzero coefficients, and the false negative rate, i.e. the proportion of important variables that are zeroed out. For $p = 20$ and $p = 50$, the standard generalized linear model approach was significantly inferior to the other seven approaches in terms of root mean squared error. For $p = 105$, where $p > n$, the generalized linear models could not be fitted, while the other methods still performed well. The double Dantzig, ridge Dantzig and interpolated Dantzig methods had low false positive rates and essentially zero false negative rates. The Gauss method had slightly lower false positive rates but correspondingly higher false negative rates. Park & Hastie's method and the generalized Dantzig selector both had extremely high false positive rates, reflecting the fact that they must include a number of incorrect variables to avoid overshrinking the coefficient estimates.

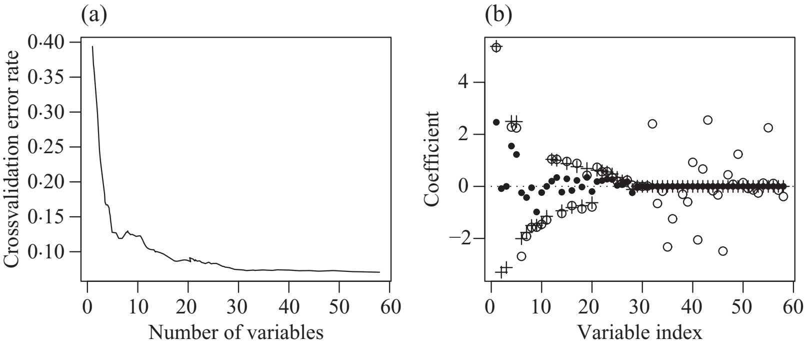

Fig. 4. Spam data. (a) Crossvalidated error rates for the double Dantzig method using different numbers of variables. (b) Coefficient estimates, ordered from largest to smallest in absolute value. The open circles correspond to generalized linear models, the closed circles to the generalized Dantzig selector and the pluses to the double Dantzig method.

5. Applications

We illustrate the performance of our methods on two real-world datasets. Given the similarities in the simulation performance of the double Dantzig, ridge Dantzig and interpolated Dantzig methods, we would expect similar results from them all and we opted to use the double Dantzig method. We also show generalized Dantzig selector results for comparison. The first dataset is the spam data described in Hastie et al. (2001), which consists of $n = 4601$ emails. For each email the response variable records whether this is a true email or a spam. We also have $p = 57$ predictors recording the relative frequencies of commonly occurring words and characters. The aim is to treat this as a two-class classification problem using the 57 predictors to identify the type of email. We began by computing the 10-fold double Dantzig crossvalidated error rate using the logistic link function on a fine grid of values for λ. We fixed ϕ at an appropriate small value. The crossvalidation required little computational effort because the generalized Dantzig selector and double Dantzig methods ran extremely quickly on datasets of this size. For each value of λ we computed the average number of variables included in the model. Figure 4(a) plots the error rate against the number of variables. Clearly, the first five to ten variables cause a significant decline in the error rate, and from that point on there is a slow decline until approximately the 30-variable model, at which point the error rate levels out.

Figure 4(b) provides a comparison of the coefficients from the 30-variable generalized Dantzig selector and double Dantzig fits to those using the standard generalized linear models. The generalized linear models coefficients are extremely variable with several that have not been plotted on this scale because they range between -45 and 5. Of the remainder, interestingly, a number of the nonzero double Dantzig coefficients closely match the generalized linear models coefficients. The double Dantzig method appears to zero-out the less important variables and to shrink the extreme coefficient estimates. The generalized Dantzig selector generally returns coefficients with the same sign as the double Dantzig but with substantial shrinkage.

Most of the reduction in error rate comes from the first few variables. In Table 2, we have listed the first 12 variables, in the order in which they appear in the model, along with their double Dantzig coefficients. Negative coefficients correspond to a decrease in the probability of spam, and positive coefficients correspond to an increase. The emails were sent to George Forman at Hewlett–Packard. Hence words such as ‘hp’, ‘hp1’ and ‘george’ are most likely indicators of non-spam, while ‘000’ and ‘$’ suggest spam. Thus all the coefficients appear to have the

Table 2. *Spam data. Coefficients and* 95% *bootstrap confidence intervals for the first* 12 *variables to enter the model when fitting the spam data. The variables are listed in the order that they entered the model*

Order	Variable	Coefficient	Confidence interval
1	hp	−1·69	(−2·67, −1·22)
2	000	3·60	(2·25, 5·54)
3	remove	3·41	(2·36, 5·36)
4	hp1	−1·54	(−2·26, −0·67)
5	$	7·74	(4·14, 10·76)
6	george	−3·10	(−5·69, −2·36)
7	1999	−0·29	(−1·26, 0·02)
8	Intercept	−0·95	(−1·78, −0·70)
9	your	0·37	(0·00, 0·51)
10	free	1·28	(0·00, 1·69)
11	edu	−1·89	(−3·09, 0·00)
12	re	−0·67	(−1·01, 0·00)

correct sign. In addition, we used the bootstrap to produce 95% percentile method confidence intervals (Efron & Tibshirani, 1993). The first six variables and intercept provide clear evidence of predictive power. The other variables include a point mass at zero so their predictive power is less clear. However, with the exception of ‘1999’, none of these variables contains values on either side of zero, which one might expect for independent variables.

The second dataset that we examined was the internet advertising data available at the University of California-Irvine machine learning repository. These data provide instances of possible advertisements on Internet pages. Hence the response is categorical, indicating whether the image is an advertisement or not. The predictors record the geometry of the image as well as phrases occurring in the URL, the image’s URL and alt text, the anchor text and words occurring near the anchor text. After pre-processing, the dataset contained $n = 2359$ observations and $p = 1430$ variables. We included this dataset because its large number of predictors presents significant statistical and computational difficulties for standard approaches. For example, the generalized linear models function of R took almost 15 minutes to run and most coefficients produced estimates that were not available. However, for the 10-variable model represented in Table 3, the generalized Dantzig selector and double Dantzig methods were able to estimate all the coefficients in under two minutes. Table 3 presents the coefficient estimates and bootstrap confidence intervals for the first ten variables to enter the model. The most important variables, in addition to the intercept, were whether or not the anchor URL contained the words ‘com’ or ‘click’ and whether or not the image URL contained the word ‘ads’.

6. Discussion

We see several possible applications of this work. For example, in an unpublished technical report for the University of Southern California, G. M. James, J. Wang & J. Zhu develop an approach to fit functional generalized linear models where the predictor, $X_i(t)$, is a function and is related to the scalar response, Y_i, though

$$g(\mu_i) = \beta_0 + \int X_i(t)\beta(t)\,dt,$$

Table 3. *Advertising data. Coefficients and* 95% *bootstrap confidence intervals for the first* 10 *variables to enter the model. The variables are listed in the order that they entered the model*

Order	Variable	Coefficient	Confidence interval
1	intercept	−3·11	(−3·46, −2·82)
2	anchor url contains com?	3·69	(2·57, 5·25)
3	image url contains ads?	3·57	(0·94, 5·52)
4	anchor url contains click?	3·45	(0·79, 5·55)
5	alt text contains click and here?	2·61	(0·00, 3·19)
6	anchor url contains uk?	2·47	(0·00, 7·89)
7	anchor url contains adclick?	6·84	(0·00, 7·71)
8	anchor url contains http or www?	4·15	(0·00, 7·24)
9	image url contains ad?	3·57	(0·00, 6·58)
10	anchor url contains cat?	2·31	(0·00, 4·13)

where $\mu_i = EY_i$. This approach estimates the coefficient curve $\beta(t)$ by assuming sparsity, not in the coefficients, but instead in the derivatives of $\beta(t)$. For example, assuming that $\beta''(t) = 0$ at most time-points produces an estimate for $\beta(t)$ that is piecewise linear with the breakpoints determined by the data. The approach uses the Dantzig selector to determine the locations of the nonzero derivatives and hence estimate $\beta(t)$. James, Wang & Zhu show that the Dantzig selector's theoretical bounds also apply here. One potential limitation with this approach is that the Dantzig selector can overshrink the derivative estimates, and the methodology suggested in this paper could be used to eliminate this shrinkage problem and produce more accurate estimates for $\beta(t)$.

Acknowledgement

We thank Professor D. M. Titterington and the referees for many helpful suggestions that improved the paper. This work was partially supported by a grant from the U.S. National Science Foundation.

Appendix

Individual steps of the generalized Dantzig selector path algorithm

Direction step. For simplicity of the notation, we suppress the superscript l throughout the remainder of the section. Compute X^* and Z^* from § 3·1, based on the latest path point, β. Our direction step and that for the DASSO are identical except that ours is applied to X^* and Z^* rather than X and Y. Let $X^*_{\mathcal{A}}$ represent the K columns of X^* corresponding to the elements of the active set $\mathcal{A}$. The set-up of the algorithm ensures that, except for some degenerate cases, the number of nonzero elements of β is either K or $K-1$. Consider the latter case first. Let d be a vector with 1s for the first $2p$ elements and 0s for the remaining K elements. Furthermore, let $\beta^+ > 0$ and $\beta^- > 0$ represent the positive and negative parts of β, i.e. $\beta = \beta^+ - \beta^-$. Finally, let S represent a K-dimensional diagonal matrix whose jth diagonal element is 1 if the jth component of $X^{\mathrm{T}}_{\mathcal{A}}(Y-\mu)$ is positive and -1 otherwise. The procedure for choosing the direction h can be summarized in the following steps.

Step 1. Compute the K by $2p+K$ matrix $A = [\,-SX^{*\mathrm{T}}_{\mathcal{A}}X^* \;\; SX^{*\mathrm{T}}_{\mathcal{A}}X^* \;\; I\,]$, where I is the identity matrix. The first p columns correspond to β^+ and the next p columns to β^-.

Step 2. We wish to identify a subset of the columns of A to form a K by K matrix B. We will construct B in the form $[\tilde{B}\;\; A_i]$, where $\tilde{B}$ is produced by selecting all the columns of A that correspond to the nonzero components of β^{+} and β^{-}, and A_i is one of the remaining columns. Write the two parts of B in the following block form:

$$\tilde{B} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \qquad A_i = \begin{pmatrix} A_{i_1} \\ A_{i_2} \end{pmatrix},$$

where B_1 is a square matrix of dimension $K - 1$, and A_{i_2} is a scalar. Finally, let $d_{\tilde{B}}$ and d_i be the components of d corresponding to the columns in $\tilde{B}$ and the column A_i, respectively.

Step 3. Choose the final column for B as A_j,

$$j = \operatorname*{arg\,max}_{i:\,|q_i| \mp 0,\, \alpha/q_i > 0} \left[d_{\tilde{B}}^{\mathrm{T}} B_1^{-1} A_{i_1} - d_i\right] / |q_i|,$$

where $q_i = A_{i_2} - B_2 B_1^{-1} A_{i_1}$, $\alpha = B_2 B_1^{-1} 1 - 1$.

Step 4. Compute $h^* = -B^{-1} 1$ and let h be a p-dimensional vector of 0s. Depending on the columns chosen for B, certain elements of h^* will correspond to elements of β^{+} and β^{-}. For each k, set $h_j = h_k^*$ if h_k^* corresponds to β_j^{+}, and set $h_j = -h_k^*$ if h_k^* corresponds to β_j^{-}. Note that h has at most K nonzero components.

In the remaining case of exactly K nonzero β-coefficients, the above procedure is simplified by removing Steps 2 and 3, and using the matrix B that is produced by selecting all the columns of A that correspond to the nonzero components of β^{+} and β^{-}; see James et al. (2009) for further explanation of, and motivation for, the direction step.

Distance step. We are looking for the next point on the path at which either a new variable becomes as correlated with the residuals as those in the active set or one of the coefficients hits zero. We start by taking the distance step of the DASSO algorithm, applied to X^* and Z^*. If the index j in Step 3 within the direction step was chosen as $2p + m$ for some positive m, remove the mth active constraint from the index set $\mathcal{A}$. Let X_k^* represent any variable that is a member of the current active set and define

$$\gamma_1 = \min_{j \in \mathcal{A}^c}{}^{+} \left\{ \frac{c_k - c_j}{(X_k^* - X_j^*)^{\mathrm{T}} X^* h}, \frac{c_k + c_j}{(X_k^* + X_j^*)^{\mathrm{T}} X^* h} \right\},$$

where $\min^{+}$ is the minimum taken over the positive components only and c_k and c_j are the kth and jth components of c. Set $\gamma_2 = \min_j^{+} \{-\beta_j / h_j\}$ and $\gamma = \min\{\gamma_1, \gamma_2\}$. Let $\tilde{\mu}$ be the mean vector corresponding to $\tilde{\beta} = \beta + \gamma h$. Since the generalized Dantzig selector path is not guaranteed to be exactly piecewise linear, it is possible that our candidate γ is too large. If this is the case we set $\gamma \leftarrow \gamma/2$, recompute $\tilde{\mu}$ and iterate until $\max_{j \in \mathcal{A}^c} |X_j^{\mathrm{T}}(Y - \tilde{\mu})| \leqslant \max_{j \in \mathcal{A}} |X_j^{\mathrm{T}}(Y - \tilde{\mu})|$.

Correction step. Define $C' = ||X^{\mathrm{T}}(Y - \tilde{\mu})||_\infty$ and compute X^* and Z^* that correspond to $\tilde{\beta}$. Form the set $\mathcal{I}$ by combining the index sets of nonzero coefficients of vectors β and $\tilde{\beta}$. Define $B = -S X_{\mathcal{A}}^{*\mathrm{T}} X_{\mathcal{I}}^*$ using S from the direction step, set $\beta_{\mathcal{I}} = B^{-1}(C' 1 - S X_{\mathcal{A}}^{*\mathrm{T}} Z^*)$, and then recompute X^*, Z^* and B. Iterate these steps until convergence, and then let β be the p-dimensional vector whose only nonzero coefficients are in the positions specified by $\mathcal{I}$ and have the values given by $\beta_{\mathcal{I}}$. Typically, very few iterations are required for convergence. Note that β satisfies $X_{\mathcal{A}}^{\mathrm{T}}(Y - \mu) = S C' 1$, so that all the covariances within the active set are now equal in magnitude.

REFERENCES

CANDÈS, E. & TAO, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n (with discussion). *Ann. Statist.* **35**, 2313–51.

CHEN, S., DONOHO, D. & SAUNDERS, M. (1998). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.* **20**, 33–61.

EFRON, B., HASTIE, T., JOHNSTON, I. & TIBSHIRANI, R. (2004). Least angle regression (with discussion). *Ann. Statist.* **32**, 407–51.

EFRON, B., HASTIE, T. & TIBSHIRANI, R. (2007). Discussion of the "Dantzig selector". *Ann. Statist.* **35**, 2358–64.

EFRON, B. & TIBSHIRANI, R. (1993) . *An Introduction to the Bootstrap*. London: Chapman & Hall.

FAN, J. & LI, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Assoc.* **96**, 1348–60.

GENKIN, A., LEWIS, D. & MADIGAN, D. (2006). Large-scale bayesian logistic regression for text categorization. *Technometrics* **49**, 291–304.

HASTIE, T. J., TIBSHIRANI, R. J. & FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. New York: Springer.

JAMES, G. M., RADCHENKO, P. & LV, J. (2009). DASSO: connections between the dantzig selector and lasso. *J. R. Statist. Soc.* B **71**, 127–142.

MCCULLAGH, P. & NELDER, J. (1989). *Generalized Linear Models*, 2nd ed. London: Chapman & Hall.

MEINSHAUSEN, N. (2007). Relaxed lasso. *Comp. Statist. Data Anal.* **52**, 374–93.

MEINSHAUSEN, N., ROCHA, G. & YU, B. (2008). A tale of three cousins: Lasso, l2 boosting and Dantzig; a discussion on the Dantzig selector: statistical estimation when p is much larger than n. *Ann. Statist.* **35**, 2373–84.

PARK, M. & HASTIE, T. (2007). An L_1-regulation path algorithm for generalized linear models. *J. R. Statist. Soc.* B **69**, 659–77.

RADCHENKO, P. & JAMES, G. M. (2008). Variable inclusion and shrinkage algorithms. *J. Am. Statist. Assoc.* **103**, 1304–15.

SHEVADE, S. & KEERTHI, S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics* **19**, 2246–53.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B **58**, 267–88.

ZHAO, P. & YU, B. (2007). Stagewise Lasso. *J. Mach. Learn. Res*. **8**, 2701–26.

ZOU, H. (2006). The adaptive lasso and its oracle properties. *J. Am. Statist. Assoc.* **101**, 1418–29.

ZOU, H. & HASTIE, T. (2005). Regularization and variable selection via the elastic net. *J. R. Statist. Soc.* B **67**, 301–20.

[*Received July* 2007. *Revised August* 2008]