Supplementary materials for this article are available online.
Please go to www.tandfonline.com/r/JCGS

A Path Algorithm for Constrained Estimation

Hua ZHOU and Kenneth LANGE

Many least-square problems involve affine equality and inequality constraints. Although there are a variety of methods for solving such problems, most statisticians find constrained estimation challenging. The current article proposes a new path-following algorithm for quadratic programming that replaces hard constraints by what are called exact penalties. Similar penalties arise in l_1 regularization in model selection. In the regularization setting, penalties encapsulate prior knowledge, and penalized parameter estimates represent a trade-off between the observed data and the prior knowledge. Classical penalty methods of optimization, such as the quadratic penalty method, solve a sequence of unconstrained problems that put greater and greater stress on meeting the constraints. In the limit as the penalty constant tends to ∞, one recovers the constrained solution. In the exact penalty method, squared penalties are replaced by absolute value penalties, and the solution is recovered for a finite value of the penalty constant. The exact path-following method starts at the unconstrained solution and follows the solution path as the penalty constant increases. In the process, the solution path hits, slides along, and exits from the various constraints. Path following in Lasso penalized regression, in contrast, starts with a large value of the penalty constant and works its way downward. In both settings, inspection of the entire solution path is revealing. Just as with the Lasso and generalized Lasso, it is possible to plot the effective degrees of freedom along the solution path. For a strictly convex quadratic program, the exact penalty algorithm can be framed entirely in terms of the sweep operator of regression analysis. A few well-chosen examples illustrate the mechanics and potential of path following. This article has supplementary materials available online.

Key Words: Exact penalty; l_1 regularization; Shape-restricted regression.

1. INTRODUCTION

When constraints appear in maximum likelihood or least-square estimation, statisticians typically resort to sophisticated commercial software or craft specific optimization algorithms for specific problems. The current article presents a new technique for solving such problems that is motivated by path following in ℓ_1 regularized regression. In penalized regression, absolute value penalties guide the trade-off in parameter estimation between the observed data and prior knowledge. Running an estimation algorithm on a grid of tuning

Hua Zhou is Assistant Professor, Department of Statistics, North Carolina State University, Raleigh, NC 27695-8203 (E-mail: *hua_zhou@ncsu.edu*). Kenneth Lange is Professor, Departments of Biomathematics, Human Genetics, and Statistics, University of California, Los Angeles, CA 90095-8076 (E-mail: *klange@ucla.edu*).

Journal of Computational and Graphical Statistics, Volume 22, Number 2, Pages 261–283
DOI: 10.1080/10618600.2012.681248

constants tends to miss important events along a path. In ℓ_1 penalized linear regression, the solution path is piecewise linear and can be anticipated. It turns out that similar considerations apply to quadratic programming with affine equality and inequality constraints. The exact penalty method of optimization replaces hard constraints by absolute value and hinge penalties and tracks the solution vector as the penalty tuning constant increases. For some finite value of the tuning constant, the penalized and constrained solutions coincide. In this article, we show how to track the solution path in quadratic programming. Besides providing the final constrained estimates, our new algorithm also delivers the whole solution path between the unconstrained and the constrained estimates. This is particularly helpful when the goal is to locate a solution between these two extremes based on criteria, such as prediction error in cross-validation.

In recent years, several path algorithms have been devised for specific l_1 regularized problems. In particular, a modification of the least angle regression (LARS) procedure can handle Lasso penalized regression (Efron et al. 2004). Rosset and Zhu (2007) gave sufficient conditions for a solution path to be piecewise linear and expanded its applications to a wider range of loss and penalty functions. Friedman (2008) derived a path algorithm for any objective function defined by the sum of a convex loss and a separable penalty (not necessarily convex). The separability restriction on the penalty term excludes many of the problems studied here. Tibshirani and Taylor (2011) devised a path algorithm for generalized Lasso problems. Their formulation is similar to ours with two differences. First, they excluded inequality constraints. Our new path algorithm handles both equality and inequality constraints gracefully. Second, they passed to the dual problem and then translated the solution path of the dual problem back to the solution path of the primal problem. We attack the primal problem directly via a simple algorithm entirely driven by the classical sweep operator of regression analysis. In our opinion, primal path following is conceptually simpler and easier to program than dual path following. Readers adept in duality theory may disagree. On the other hand, the dual approach makes fewer restrictions on constraint gradients and can, in principle, deal with a wider variety of equality-constrained problems. The degrees of freedom formula derived for the Lasso (Efron et al. 2004; Zou, Hastie, and Tibshirani 2007) and generalized Lasso (Tibshirani and Taylor 2011) apply equally well in the presence of inequality constraints.

Our object of study will be minimization of the quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^t\boldsymbol{x} + c, \tag{1}$$

subject to the affine equality constraints $\boldsymbol{V}\boldsymbol{x} = \boldsymbol{d}$ and the affine inequality constraints $\boldsymbol{W}\boldsymbol{x} \leq \boldsymbol{e}$.Throughout our discussion, we assume that the feasible region is nontrivial and that the minimum is attained. If the symmetric matrix $\boldsymbol{A}$ has a negative eigenvalue λ and corresponding unit eigenvector $\boldsymbol{u}$, then $\lim_{r\to\infty} f(r\boldsymbol{u}) = -\infty$ because the quadratic term $\frac{1}{2}(r\boldsymbol{u})^t\boldsymbol{A}(r\boldsymbol{u}) = \frac{\lambda}{2}r^2$ dominates the linear term $r\boldsymbol{b}^t\boldsymbol{u}$. To avoid such behavior, we initially assume that all eigenvalues of $\boldsymbol{A}$ are positive. This makes $f(\boldsymbol{x})$ strictly convex and coercive and guarantees a unique minimum point subject to the constraints. In linear regression, $\boldsymbol{A} = \boldsymbol{X}^t\boldsymbol{X}$ for some design matrix $\boldsymbol{X}$. In this setting, $\boldsymbol{A}$ is positive definite, provided $\boldsymbol{X}$ has full column rank. The latter condition is only possible when the number of cases equals or exceeds the number of predictors. If $\boldsymbol{A}$ is positive semidefinite and singular, then adding a

small amount of ridge regularization $\epsilon \boldsymbol{I}$ to it can be helpful (Tibshirani and Taylor 2011). Later we indicate how path following extends to positive semidefinite or even indefinite matrices $\boldsymbol{A}$. Our assumption that the rows of $\boldsymbol{V}$ and $\boldsymbol{W}$ are linearly independent excludes problems such as the sparse fused Lasso and two- and three-dimensional fused Lasso considered by Tibshirani and Taylor (2011). We discuss the difficulties in relaxing this assumption in Section 5 and suggest a numerical remedy.

In multitask learning, the response is a d-dimensional vector $\boldsymbol{Y} \in \mathbb{R}^d$, and one minimizes the squared Frobenius deviation

$$\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_{\mathrm{F}}^2 \tag{2}$$

with respect to the $p \times d$ regression coefficient matrix $\boldsymbol{B}$. When the constraints take the form $\boldsymbol{V}\boldsymbol{B} \leq \boldsymbol{D}$ and $\boldsymbol{W}\boldsymbol{B} = \boldsymbol{E}$, the problem reduces to quadratic programming as just posed. Indeed, if we stack the columns of $\boldsymbol{Y}$ with the vec operator, then the problem involves minimizing $\frac{1}{2}\|\mathrm{vec}(\boldsymbol{Y}) - (\boldsymbol{I} \otimes \boldsymbol{X})\mathrm{vec}(\boldsymbol{B})\|_2^2$. Here, the identity $\mathrm{vec}(\boldsymbol{X}\boldsymbol{B}) = (\boldsymbol{I} \otimes \boldsymbol{X})\mathrm{vec}(\boldsymbol{B})$ comes into play invoking the Kronecker product and the identity matrix $\boldsymbol{I}$. Similarly, we can rewrite the constraints as $(\boldsymbol{I} \otimes \boldsymbol{V})\mathrm{vec}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{D})$ and $(\boldsymbol{I} \otimes \boldsymbol{W})\mathrm{vec}(\boldsymbol{X}) \leq \mathrm{vec}(\boldsymbol{E})$.

As an illustration, consider the classical concave regression problem (Hildreth 1954). The data consist of a scatterplot (x_i, y_i) of n points with associated weights w_i and predictors x_i arranged in increasing order. The concave regression problem seeks the estimates θ_i that minimize the weighted sum of squares

$$\sum_{i=1}^{n} w_i(y_i - \theta_i)^2 \tag{3}$$

subject to the concavity constraints

$$\frac{\theta_i - \theta_{i-1}}{x_i - x_{i-1}} \geq \frac{\theta_{i+1} - \theta_i}{x_{i+1} - x_i}, \quad i = 2, \ldots, n-1. \tag{4}$$

The consistency of concave regression is proved by Hanson and Pledger (1976); the asymptotic distribution of the estimates and their rate of convergence are studied in subsequent articles (Mammen 1991; Groeneboom, Jongbloed, and Wellner 2001). Figure 1 shows a scatterplot of 100 data points. Here, the x_i are uniformly sampled from the interval [0,1], the weights are constant, and $y_i = 4x_i(1 - x_i) + \epsilon_i$, where the ϵ_i are iid normal with mean 0 and standard deviation $\sigma = 0.3$. The left panel of Figure 1 gives four snapshots of the solution path. The original data points $\hat{\theta}_i = y_i$ provide the unconstrained estimates. The solid line shows the concavity-constrained solution. The dotted and dashed lines represent intermediate solutions between the unconstrained and the constrained solutions. The degrees of freedom formula derived in Section 6 is a vehicle for model selection based on criterion such as C_p, the Akaike information criterion (AIC), and the Bayesian information criterion (BIC). For example, the C_p statistic

$$C_p(\hat{\boldsymbol{\theta}}) = \frac{1}{n}\|\boldsymbol{y} - \hat{\boldsymbol{\theta}}\|_2^2 + \frac{2}{n}\sigma^2 \mathrm{df}(\hat{\boldsymbol{\theta}})$$

is an unbiased estimator of the true prediction error (Efron 2004) under the estimator $\hat{\boldsymbol{\theta}}$ whenever an unbiased estimate of the degrees of freedom is used. The right panel shows the C_p statistic along the solution path. In this example, the design matrix is a diagonal

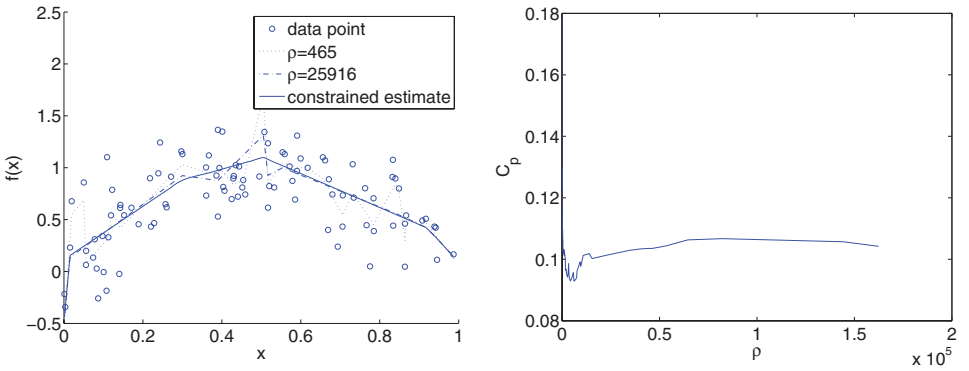

Figure 1. Path solutions to the concave regression problem. Left: the unconstrained solution (original data points), two intermediate solutions (dotted and dashed lines), and the concavity-constrained solution (solid line). Right: the C_p statistic as a function of the penalty constant ρ along the solution path. The online version of this figure is in color.

matrix. After submitting this article, we learned that Tibshirani, Hoefling, and Tibshirani (2011) solved a similar convex regression problem by a path algorithm. As we will see in Section 7, postulating a more general design matrix or other kinds of constraints broadens the scope of applications of the path algorithm and the estimated degrees of freedom.

Here is a road map to the remainder of the current article. Section 2 reviews the exact penalty method for optimization and clarifies the connections between constrained optimization and regularization in statistics. Section 3 derives in detail our path algorithm. Its implementation via the sweep operator and QR decomposition are described in Sections 4 and 5. Section 6 derives the degrees of freedom formula. Section 7 presents various numerical examples. Finally, Section 8 discusses the limitations of the path algorithm and hints at future generalizations.

2. THE EXACT PENALTY METHOD

Exact penalty methods minimize the function

$$\mathcal{E}_\rho(\boldsymbol{x}) = f(\boldsymbol{x}) + \rho \sum_{i=1}^{r} |g_i(\boldsymbol{x})| + \rho \sum_{j=1}^{s} \max\{0, h_j(\boldsymbol{x})\},$$

where $f(\boldsymbol{x})$ is the objective function, $g_i(\boldsymbol{x}) = 0$ is one of r equality constraints, and $h_j(\boldsymbol{x}) \le 0$ is one of s inequality constraints. It is interesting to compare this function with the Lagrangian function

$$\mathcal{L}(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{s} \mu_j h_j(\boldsymbol{x})$$

that captures the behavior of $f(\boldsymbol{x})$ at a constrained local minimum $\boldsymbol{y}$. By definition, the Lagrange multipliers satisfy the conditions $\nabla \mathcal{L}(\boldsymbol{y}) = \boldsymbol{0}$ and $\mu_j \ge 0$ and $\mu_j h_j(\boldsymbol{y}) = 0$ for

all j. In the exact penalty method, one takes

$$\rho > \max\{|\lambda_1|, \ldots, |\lambda_r|, \mu_1, \ldots, \mu_s\}. \tag{5}$$

This choice creates the majorization $f(\boldsymbol{x}) \leq \mathcal{E}_\rho(\boldsymbol{x})$ with $f(\boldsymbol{z}) = \mathcal{E}_\rho(\boldsymbol{z})$ at any feasible point $\boldsymbol{z}$. Thus, minimizing $\mathcal{E}_\rho(\boldsymbol{x})$ forces $f(\boldsymbol{x})$ downhill. Much more than this is going on, however. As the next proposition proves, minimizing $\mathcal{E}_\rho(\boldsymbol{x})$ effectively minimizes $f(\boldsymbol{x})$ subject to the constraints.

Proposition 1. Suppose the objective function $f(\boldsymbol{x})$ and the constraint functions are twice differentiable and satisfy the Lagrange multiplier rule at the local minimum $\boldsymbol{y}$. If inequality (5) holds and $\boldsymbol{v}^* d^2\mathcal{L}(\boldsymbol{y})\boldsymbol{v} > 0$ for every vector $\boldsymbol{v} \neq \boldsymbol{0}$ satisfying $dg_i(\boldsymbol{y})\boldsymbol{v} = 0$ and $dh_j(\boldsymbol{y})\boldsymbol{v} \leq 0$ for all active inequality constraints, then $\boldsymbol{y}$ furnishes an unconstrained local minimum of $\mathcal{E}_\rho(\boldsymbol{x})$. If $f(\boldsymbol{x})$ is convex, the $g_i(\boldsymbol{x})$ are affine, the $h_j(\boldsymbol{x})$ are convex, and Slater's constraint qualification holds, then $\boldsymbol{y}$ is a minimum of $\mathcal{E}_\rho(\boldsymbol{x})$ if and only if $\boldsymbol{y}$ is a minimum of $f(\boldsymbol{x})$ subject to the constraints. In this convex programming context, no differentiability assumptions are needed.

Proof. The conditions imposed on the quadratic form $\boldsymbol{v}^* d^2\mathcal{L}(\boldsymbol{y})\boldsymbol{v} > 0$ are well-known sufficient conditions for a local minimum. Theorems 6.9 and 7.21 of Ruszczyński (2006) prove all of the foregoing assertions. □

3. THE PATH-FOLLOWING ALGORITHM

In the quadratic programming context with objective function (1), affine equality constraints $\boldsymbol{V}\boldsymbol{x} = \boldsymbol{d}$, and affine inequality constraints $\boldsymbol{W}\boldsymbol{x} \leq \boldsymbol{e}$, the penalized objective function takes the form

$$\mathcal{E}_\rho(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^t\boldsymbol{x} + c + \rho\sum_{i=1}^{r}\left|\boldsymbol{v}_i^t\boldsymbol{x} - d_i\right| + \rho\sum_{j=1}^{s}\left(\boldsymbol{w}_j^t\boldsymbol{x} - e_i\right)_+. \tag{6}$$

Our assumptions on $\boldsymbol{A}$ render $\mathcal{E}_\rho(\boldsymbol{x})$ strictly convex and coercive and guarantee a unique minimum point $\boldsymbol{x}(\rho)$. The generalized Lasso problem studied by Tibshirani and Taylor (2011) drops the last term and consequently excludes inequality-constrained applications.

According to the rules of the convex calculus (Ruszczyński 2006), the unique optimal point $\boldsymbol{x}(\rho)$ of the function $\mathcal{E}_\rho(\boldsymbol{x})$ is characterized by the stationarity condition

$$\boldsymbol{0} = \boldsymbol{A}\boldsymbol{x}(\rho) + \boldsymbol{b} + \rho\sum_{i=1}^{r} s_i(\rho)\boldsymbol{v}_i + \rho\sum_{j=1}^{s} t_j(\rho)\boldsymbol{w}_j, \tag{7}$$

with coefficients

$$s_i(\rho) \in \begin{cases} \{-1\} & \boldsymbol{v}_i^t\boldsymbol{x}(\rho) - d_i < 0, \\ [-1, 1] & \boldsymbol{v}_i^t\boldsymbol{x}(\rho) - d_i = 0, \\ \{1\} & \boldsymbol{v}_i^t\boldsymbol{x}(\rho) - d_i > 0, \end{cases} \qquad t_j(\rho) \in \begin{cases} \{0\} & \boldsymbol{w}_j^t\boldsymbol{x}(\rho) - e_i < 0, \\ [0, 1] & \boldsymbol{w}_j^t\boldsymbol{x}(\rho) - e_i = 0, \\ \{1\} & \boldsymbol{w}_j^t\boldsymbol{x}(\rho) - e_i > 0. \end{cases} \tag{8}$$

Assuming that the vectors $(\cup_i\{\boldsymbol{v}_i\}) \cup (\cup_j\{\boldsymbol{w}_j\})$ are linearly independent, the coefficients $s_i(\rho)$ and $t_j(\rho)$ are uniquely determined. The sets defining the possible values of $s_i(\rho)$ and

$t_j(\rho)$ are the subdifferentials of the functions $|s_i(\rho)|$ and $t_j(\rho)_+ = \max\{0, t_j(\rho)\}$. The coefficients s_i and t_j appear as the dual variables in the dual path algorithm of Tibshirani and Taylor (2011). We now prove that the solution and coefficient paths are continuous.

Proposition 2. If $\boldsymbol{A}$ is positive definite and the vectors $(\cup_i\{\boldsymbol{v}_i\}) \cup (\cup_j\{\boldsymbol{w}_j\})$ are linearly independent, then the solution path $\boldsymbol{x}(\rho)$ and the coefficient paths $\boldsymbol{s}(\rho)$ and $\boldsymbol{t}(\rho)$ are unique and continuous.

Proof. The representation

$$\boldsymbol{x}(\rho) = -\boldsymbol{A}^{-1}\left(\boldsymbol{b} + \rho\sum_{i=1}^{r} s_i(\rho)\boldsymbol{v}_i + \rho\sum_{j=1}^{s} t_j(\rho)\boldsymbol{w}_j\right)$$

entails the norm inequality

$$\|\boldsymbol{x}(\rho)\| \le \|\boldsymbol{A}^{-1}\|\left(\|\boldsymbol{b}\| + \rho\sum_{i=1}^{r}\|\boldsymbol{v}_i\| + \rho\sum_{j=1}^{s}\|\boldsymbol{w}_j\|\right).$$

Thus, the solution vector $\boldsymbol{x}(\rho)$ is bounded whenever $\rho \ge 0$ is bounded above. To prove continuity, suppose that it fails for a given ρ. Then, there exists an $\epsilon > 0$ and a sequence ρ_n tending to ρ such that $\|\boldsymbol{x}(\rho_n) - \boldsymbol{x}(\rho)\| \ge \epsilon$ for all n. Since $\boldsymbol{x}(\rho_n)$ is bounded, we can pass to a subsequence if necessary and assume that $\boldsymbol{x}(\rho_n)$ converges to some point $\boldsymbol{y}$. Taking limits in the inequality $\mathcal{E}_{\rho_n}[\boldsymbol{x}(\rho_n)] \le \mathcal{E}_{\rho_n}(\boldsymbol{x})$ demonstrates that $\mathcal{E}_\rho(\boldsymbol{y}) \le \mathcal{E}_\rho(\boldsymbol{x})$ for all $\boldsymbol{x}$. Because $\boldsymbol{x}(\rho)$ is unique, we reach the contradictory conclusions $\|\boldsymbol{y} - \boldsymbol{x}(\rho)\| \ge \epsilon$ and $\boldsymbol{y} = \boldsymbol{x}(\rho)$. Continuity is inherited by the coefficients $s_i(\rho)$ and $t_j(\rho)$. Indeed, let $\boldsymbol{V}$ and $\boldsymbol{W}$ be the matrices with rows $\boldsymbol{v}_i^t$ and $\boldsymbol{w}_j^t$, and let $\boldsymbol{U}$ be the block matrix $\binom{V}{W}$. The stationarity condition can be restated as

$$\boldsymbol{0} = \boldsymbol{A}\boldsymbol{x}(\rho) + \boldsymbol{b} + \rho\boldsymbol{U}^t\begin{pmatrix}\boldsymbol{s}(\rho)\\ \boldsymbol{t}(\rho)\end{pmatrix}.$$

Multiplying this equation by $\boldsymbol{U}$ and solving give

$$\rho\begin{pmatrix}\boldsymbol{s}(\rho)\\ \boldsymbol{t}(\rho)\end{pmatrix} = -(\boldsymbol{U}\boldsymbol{U}^t)^{-1}\boldsymbol{U}[\boldsymbol{A}\boldsymbol{x}(\rho) + \boldsymbol{b}], \tag{9}$$

and the continuity of the left-hand side follows from the continuity of $\boldsymbol{x}(\rho)$. Finally, dividing by ρ yields the continuity of the coefficients $s_i(\rho)$ and $t_j(\rho)$ for $\rho > 0$. □

Positive definiteness of $\boldsymbol{A}$ is not required for the uniqueness of $\boldsymbol{x}(\rho)$. The penalized objective function (6) may have a unique minimum for large ρ even when $\boldsymbol{A}$ is not positive definite. In our subsequent derivation of the path algorithm, we will also observe that the uniqueness of the coefficient paths $\boldsymbol{s}(\rho)$ and $\boldsymbol{t}(\rho)$ only requires linear independence of the active constraints along the solution path. In this and the next section, we assume strict convexity of $\boldsymbol{A}$ and linear independence of all constraint vectors $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$. In Section 5, we discuss extensions of the path algorithm where the first restriction is relaxed.

We next show that the solution path is piecewise linear. Along the path, we keep track of the following index sets determined by the constraint residuals:

$$\mathcal{N}_{\mathrm{E}} = \{i : \boldsymbol{v}_i^t \boldsymbol{x} - d_i < 0\}, \quad \mathcal{N}_{\mathrm{I}} = \{j : \boldsymbol{w}_j^t \boldsymbol{x} - e_j < 0\},$$
$$\mathcal{Z}_{\mathrm{E}} = \{i : \boldsymbol{v}_i^t \boldsymbol{x} - d_i = 0\}, \quad \mathcal{Z}_{\mathrm{I}} = \{j : \boldsymbol{w}_j^t \boldsymbol{x} - e_j = 0\},$$
$$\mathcal{P}_{\mathrm{E}} = \{i : \boldsymbol{v}_i^t \boldsymbol{x} - d_i > 0\}, \quad \mathcal{P}_{\mathrm{I}} = \{j : \boldsymbol{w}_j^t \boldsymbol{x} - e_j > 0\}.$$

We drop the argument ρ from $\boldsymbol{x}(\rho)$ whenever notationally convenient. The reader should keep in mind that these index sets are functions of ρ as well. For the sake of simplicity, assume that at the beginning of the current segment, s_i does not equal -1 or 1 when $i \in \mathcal{Z}_{\mathrm{E}}$ and t_j does not equal 0 or 1 when $j \in \mathcal{Z}_{\mathrm{I}}$. In other words, the coefficients of the active constraints occur in the interior of their subdifferentials. Let us show in this circumstance that the solution path can be extended in a linear fashion. The general idea is to impose the equality constraints $\boldsymbol{V}_{\mathcal{Z}_{\mathrm{E}}} \boldsymbol{x} = \boldsymbol{d}_{\mathcal{Z}_{\mathrm{E}}}$ and $\boldsymbol{W}_{\mathcal{Z}_{\mathrm{I}}} \boldsymbol{x} = \boldsymbol{e}_{\mathcal{Z}_{\mathrm{I}}}$ and write the objective function $\mathcal{E}_\rho(\boldsymbol{x})$ as

$$\frac{1}{2}\boldsymbol{x}^t \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^t \boldsymbol{x} + c - \rho \sum_{i \in \mathcal{N}_{\mathrm{E}}} (\boldsymbol{v}_i^t \boldsymbol{x} - d_i) + \rho \sum_{i \in \mathcal{P}_{\mathrm{E}}} (\boldsymbol{v}_i^t \boldsymbol{x} - d_i) + \rho \sum_{j \in \mathcal{P}_{\mathrm{I}}} (\boldsymbol{w}_j^t \boldsymbol{x} - e_j).$$

For notational convenience, define

$$\boldsymbol{U}_{\mathcal{Z}} = \begin{pmatrix} \boldsymbol{V}_{\mathcal{Z}_{\mathrm{E}}} \\ \boldsymbol{W}_{\mathcal{Z}_{\mathrm{I}}} \end{pmatrix}, \quad \boldsymbol{c}_{\mathcal{Z}} = \begin{pmatrix} \boldsymbol{d}_{\mathcal{Z}_{\mathrm{E}}} \\ \boldsymbol{e}_{\mathcal{Z}_{\mathrm{I}}} \end{pmatrix}, \quad \boldsymbol{u}_{\bar{\mathcal{Z}}} = -\sum_{i \in \mathcal{N}_{\mathrm{E}}} \boldsymbol{v}_i + \sum_{i \in \mathcal{P}_{\mathrm{E}}} \boldsymbol{v}_i + \sum_{j \in \mathcal{P}_{\mathrm{I}}} \boldsymbol{w}_j.$$

Minimizing $\mathcal{E}_\rho(\boldsymbol{x})$ subject to the constraints generates the Lagrange multiplier problem

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{U}_{\mathcal{Z}}^t \\ \boldsymbol{U}_{\mathcal{Z}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda}_{\mathcal{Z}} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b} - \rho \boldsymbol{u}_{\bar{\mathcal{Z}}} \\ \boldsymbol{c}_{\mathcal{Z}} \end{pmatrix}, \tag{10}$$

with the explicit path solution and Lagrange multipliers

$$\boldsymbol{x}(\rho) = -\boldsymbol{P}(\boldsymbol{b} + \rho \boldsymbol{u}_{\bar{\mathcal{Z}}}) + \boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}} = -\rho \boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}} - \boldsymbol{P}\boldsymbol{b} + \boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}}, \tag{11}$$

$$\boldsymbol{\lambda}_{\mathcal{Z}} = -\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}} - \rho \boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}. \tag{12}$$

Here,

$$\begin{pmatrix} \boldsymbol{P} & \boldsymbol{Q} \\ \boldsymbol{Q}^t & \boldsymbol{R} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{U}_{\mathcal{Z}}^t \\ \boldsymbol{U}_{\mathcal{Z}} & \boldsymbol{0} \end{pmatrix}^{-1},$$

with

$$\boldsymbol{P} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}_{\mathcal{Z}}^t \left(\boldsymbol{U}_{\mathcal{Z}} \boldsymbol{A}^{-1} \boldsymbol{U}_{\mathcal{Z}}^t\right)^{-1} \boldsymbol{U}_{\mathcal{Z}} \boldsymbol{A}^{-1},$$
$$\boldsymbol{Q} = \boldsymbol{A}^{-1}\boldsymbol{U}_{\mathcal{Z}}^t \left(\boldsymbol{U}_{\mathcal{Z}} \boldsymbol{A}^{-1} \boldsymbol{U}_{\mathcal{Z}}^t\right)^{-1},$$
$$\boldsymbol{R} = -\left(\boldsymbol{U}_{\mathcal{Z}} \boldsymbol{A}^{-1} \boldsymbol{U}_{\mathcal{Z}}^t\right)^{-1}.$$

As we will see in the next section, these seemingly complicated objects arise naturally if path following is organized around the sweep operator.

It is clear that as we increase ρ, the solution path (11) and the multiplier path (12) change in a linear fashion until either an inactive constraint becomes active or the coefficient of an

active constraint hits the boundary of its subdifferential. We investigate the first case first. Imagining ρ to be a time parameter, an inactive constraint $i \in \mathcal{N}_{\mathrm{E}} \cup \mathcal{P}_{\mathrm{E}}$ becomes active when

$$\boldsymbol{v}_i^t \boldsymbol{x}(\rho) = -\boldsymbol{v}_i^t \boldsymbol{P}(\boldsymbol{b} + \rho \boldsymbol{u}_{\bar{\mathcal{Z}}}) + \boldsymbol{v}_i^t \boldsymbol{Q} \boldsymbol{c}_{\mathcal{Z}} = d_i.$$

If this event occurs, it occurs at the hitting time

$$\rho^{(i)} = \frac{-\boldsymbol{v}_i^t \boldsymbol{P}\boldsymbol{b} + \boldsymbol{v}_i^t \boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}} - d_i}{\boldsymbol{v}_i^t \boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}}}. \tag{13}$$

Similarly, an inactive constraint $j \in \mathcal{N}_{\mathrm{I}} \cup \mathcal{P}_{\mathrm{I}}$ becomes active at the hitting time

$$\rho^{(j)} = \frac{-\boldsymbol{w}_j^t \boldsymbol{P}\boldsymbol{b} + \boldsymbol{w}_j^t \boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}} - e_j}{\boldsymbol{w}_j^t \boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}}}. \tag{14}$$

To determine the escape time for an active constraint, consider once again the stationarity condition (7). The Lagrange multiplier corresponding to an active constraint coincides with a product $\rho s_i(\rho)$ or $\rho t_j(\rho)$. Therefore, if we collect the coefficients for the active constraints into the vector $\boldsymbol{r}_{\mathcal{Z}}(\rho)$, then Equation (12) implies

$$\boldsymbol{r}_{\mathcal{Z}}(\rho) = \frac{1}{\rho}\boldsymbol{\lambda}_{\mathcal{Z}}(\rho) = \frac{1}{\rho}(-\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}) - \boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}. \tag{15}$$

Formula (15) for $\boldsymbol{r}_{\mathcal{Z}}(\rho)$ can be rewritten in terms of the value $\boldsymbol{r}_{\mathcal{Z}}(\rho_0)$ at the start ρ_0 of the current segment as

$$\boldsymbol{r}_{\mathcal{Z}}(\rho) = \frac{\rho_0}{\rho}\boldsymbol{r}_{\mathcal{Z}}(\rho_0) - \left(1 - \frac{\rho_0}{\rho}\right) \boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}. \tag{16}$$

It is clear that $\boldsymbol{r}_{\mathcal{Z}}(\rho)_i$ is increasing in ρ when $[\boldsymbol{r}_{\mathcal{Z}}(\rho_0) + \boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}]_i < 0$ and decreasing in ρ when the reverse is true. The coefficient of an active constraint $i \in \mathcal{Z}_{\mathrm{E}}$ escapes at either of the times

$$\rho^{(i)} = \frac{[-\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}]_i}{[\boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}]_i - 1} \quad \text{or} \quad \frac{[-\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}]_i}{[\boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}]_i + 1},$$

whichever is pertinent. Similarly, the coefficient of an active constraint $j \in \mathcal{Z}_{\mathrm{I}}$ escapes at either of the times

$$\rho^{(j)} = \frac{[-\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}]_j}{[\boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}]_j} \quad \text{or} \quad \frac{[-\boldsymbol{Q}^t \boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}]_j}{[\boldsymbol{Q}^t \boldsymbol{u}_{\bar{\mathcal{Z}}}]_j + 1},$$

whichever is pertinent. The earliest hitting time or escape time over all constraints determines the duration of the current linear segment.

At the end of the current segment, our assumption that all active coefficients occur in the interior of their subdifferentials is actually violated. When the hitting time for an inactive constraint occurs first, we move the constraint to the appropriate active set $\mathcal{Z}_{\mathrm{E}}$ or $\mathcal{Z}_{\mathrm{I}}$ and keep the other constraints in place. Similarly, when the escape time for an active constraint occurs first, we move the constraint to the appropriate inactive set and keep the other constraints in place. In the second scenario, if s_i hits the value -1, then we move i to $\mathcal{N}_{\mathrm{E}}$. If s_i hits the value 1, then we move i to $\mathcal{P}_{\mathrm{E}}$. Similar comments apply when a coefficient t_j hits 0 or 1. Once this move is executed, we commence a new linear segment as just described.

The path-following algorithm continues segment by segment until for sufficiently large ρ, the sets $\mathcal{N}_{\mathrm{E}}$, $\mathcal{P}_{\mathrm{E}}$, and $\mathcal{P}_{\mathrm{I}}$ are exhausted, $\boldsymbol{u}_{\bar{\mathcal{Z}}} = \boldsymbol{0}$, and the solution vector (11) stabilizes.

This description omits two details. First, to get the process started, we set $\rho = 0$ and $\boldsymbol{x}(0) = -\boldsymbol{A}^{-1}\boldsymbol{b}$. In other words, we start at the unconstrained minimum. For inactive constraints, the coefficients $s_i(0)$ and $t_j(0)$ are fixed. However, for active constraints, it is unclear how to assign the coefficients and whether to release the constraints from active status as ρ increases. Second, very rarely, some of the hitting times and escape times will coincide. We are then faced again with the problem of which of the active constraints, with coefficients on their subdifferential boundaries, to keep active and which to encourage to go inactive in the next segment. In practice, the first problem can easily occur. Roundoff error typically keeps the second problem at bay.

In both anomalous cases, the status of each of active constraint can be resolved by trying all possibilities. Consider the second case first. If there are a currently active constraints parked at their subdifferential boundaries, then there are 2^a possible configurations for their active–inactive states in the next segment. For a given configuration, we can exploit formula (15) to check whether the coefficient for an active constraint occurs in its subdifferential. If the coefficient occurs on the boundary of its subdifferential, then we can use representation (16) to check whether it is headed into the interior of the subdifferential as ρ increases. Since the path and its coefficients are unique, one and only one configuration should determine the next linear segment. At the start of the path algorithm, the correct configuration also determines the initial values of the active coefficients. If we take limits in Equation (15) as ρ tends to 0, then the coefficients will escape their subdifferentials unless $-\boldsymbol{Q}^t\boldsymbol{b} + \boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}} = \boldsymbol{0}$ and all components of $-\boldsymbol{Q}^t\boldsymbol{u}_{\bar{\mathcal{Z}}}$ lie in their appropriate subdifferentials. Hence, again it is easy to decide on the active set $\mathcal{Z}$ going forward from $\rho = 0$. One could object that the number of configurations 2^a is potentially very large, but, in practice, this combinatorial bottleneck never occurs. Visiting the various configurations can be viewed as a systematic walk through the subsets of $\{1, \ldots, a\}$ and organized using a classical gray code (Savage 1997) that deletes at most one element and adjoins at most one element as one passes from one active subset to the next. As we will see in the next section, adjoining an element corresponds to sweeping a diagonal entry of a tableau and deleting an element corresponds to inverse sweeping a diagonal entry of the same tableau.

When a is large, a more economical solution is to minimize the penalized objective function (6) at $\rho + \epsilon$ for ϵ small using any unconstrained optimizer for nonsmooth problems. Reasonable choices include the proximal gradient method (Chen et al. 2010), Nesterov's method (Liu, Yuan, and Ye 2010), and coordinate descent after reparameterization (Friedman et al. 2007; Wu and Lange 2008). The solution initializes the set configuration at time $\rho + \epsilon$ in anticipation of the resumption of path following.

4. THE PATH ALGORITHM AND SWEEPING

Implementation of the path algorithm can be conveniently organized around the sweep and inverse sweep operators of regression analysis (Dempster 1969; Jennrich 1977; Goodnight 1979; Little and Rubin 2002; Lange 2010). We first recall the definition and basic properties of the sweep operator. Suppose $\boldsymbol{A}$ is an $m \times m$ symmetric matrix. Sweeping

on the kth diagonal entry $a_{kk} \neq 0$ of $\boldsymbol{A}$ yields a new symmetric matrix $\widehat{\boldsymbol{A}}$ with entries

$$\hat{a}_{kk} = -\frac{1}{a_{kk}}, \qquad \hat{a}_{ik} = \frac{a_{ik}}{a_{kk}}, \quad i \neq k,$$
$$\hat{a}_{kj} = \frac{a_{kj}}{a_{kk}}, \quad j \neq k, \quad \hat{a}_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}}, \quad i, j \neq k.$$

These arithmetic operations can be undone by inverse sweeping on the same diagonal entry. Inverse sweeping sends the symmetric matrix $\boldsymbol{A}$ into the symmetric matrix $\check{\boldsymbol{A}}$ with entries

$$\check{a}_{kk} = -\frac{1}{a_{kk}}, \qquad \check{a}_{ik} = -\frac{a_{ik}}{a_{kk}}, \quad i \neq k,$$
$$\check{a}_{kj} = -\frac{a_{kj}}{a_{kk}}, \quad j \neq k, \quad \check{a}_{ij} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}}, \quad i, j \neq k.$$

Both sweeping and inverse sweeping preserve symmetry. Thus, all operations can be carried out on either the lower or the upper triangle of $\boldsymbol{A}$ alone, saving both computational time and storage. When several sweeps or inverse sweeps are performed, their order is irrelevant. Finally, a symmetric matrix $\boldsymbol{A}$ is positive definite if and only if $\boldsymbol{A}$ can be completely swept, and all of its diagonal entries remain positive until swept. Complete sweeping produces $-\boldsymbol{A}^{-1}$. Each sweep of a positive definite matrix reduces the magnitude of the unswept diagonal entries. Positive definite matrices with poor condition numbers can be detected by monitoring the relative magnitude of each diagonal entry just prior to sweeping.

At the start of path following, we initialize a path tableau with block entries

$$\left(\begin{array}{c|cc} -\boldsymbol{A} & -\boldsymbol{U}^t & \boldsymbol{b} \\ \hline * & \boldsymbol{0} & -\boldsymbol{c} \\ * & * & 0 \end{array}\right). \tag{17}$$

The starred blocks here are determined by symmetry. Sweeping the diagonal entries of the upper-left block $-\boldsymbol{A}$ of the tableau yields

$$\left(\begin{array}{c|cc} \boldsymbol{A}^{-1} & \boldsymbol{A}^{-1}\boldsymbol{U}^t & -\boldsymbol{A}^{-1}\boldsymbol{b} \\ \hline * & \boldsymbol{U}\boldsymbol{A}^{-1}\boldsymbol{U}^t & -\boldsymbol{U}\boldsymbol{A}^{-1}\boldsymbol{b} - \boldsymbol{c} \\ * & * & \boldsymbol{b}^t\boldsymbol{A}^{-1}\boldsymbol{b} \end{array}\right).$$

The new tableau contains the unconstrained solution $\boldsymbol{x}(0) = -\boldsymbol{A}^{-1}\boldsymbol{b}$ and the corresponding constraint residuals $-\boldsymbol{U}\boldsymbol{A}^{-1}\boldsymbol{b} - \boldsymbol{c}$. In path following, we adopt our previous notation and divide the original tableau into subblocks. The result

$$\left(\begin{array}{cc|cc} -\boldsymbol{A} & -\boldsymbol{U}_{\mathcal{Z}}^t & -\boldsymbol{U}_{\tilde{\mathcal{Z}}}^t & \boldsymbol{b} \\ * & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{c}_{\mathcal{Z}} \\ \hline * & * & \boldsymbol{0} & -\boldsymbol{c}_{\tilde{\mathcal{Z}}} \\ * & * & * & 0 \end{array}\right) \tag{18}$$

highlights the active and inactive constraints. If we continue sweeping until all diagonal entries of the upper-left quadrant of this version of the tableau are swept, then the

tableau becomes

$$\left(\begin{array}{cc|cc} \boldsymbol{P} & \boldsymbol{Q} & \boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t & -\boldsymbol{P}\boldsymbol{b}+\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}} \\ * & \boldsymbol{R} & \boldsymbol{Q}^t\boldsymbol{U}_{\bar{\mathcal{Z}}}^t & -\boldsymbol{Q}^t\boldsymbol{b}+\boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}} \\ \hline * & * & \boldsymbol{U}_{\bar{\mathcal{Z}}}\boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t & \boldsymbol{U}_{\bar{\mathcal{Z}}}(-\boldsymbol{P}\boldsymbol{b}+\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}})-\boldsymbol{c}_{\bar{\mathcal{Z}}} \\ * & * & * & \boldsymbol{b}^t\boldsymbol{P}\boldsymbol{b}-2\boldsymbol{b}^t\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}}+\boldsymbol{c}_{\bar{\mathcal{Z}}}^t\boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}} \end{array}\right).$$

All of the required elements for the path algorithm now magically appear.

Given the next ρ, the solution vector $\boldsymbol{x}(\rho)$ appearing in Equation (11) requires the sum $-\boldsymbol{P}\boldsymbol{b}+\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}}$, which occurs in the revised tableau, and the vector $\boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}}$. If $\boldsymbol{r}_{\bar{\mathcal{Z}}}$ denotes the coefficient vector for the inactive constraints, with entries of -1 for constraints in $\mathcal{N}_{\mathrm{E}}$, 0 for constraints in $\mathcal{N}_{\mathrm{I}}$, and 1 for constraints in $\mathcal{P}_{\mathrm{E}} \cup \mathcal{P}_{\mathrm{I}}$, then $\boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}} = \boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t\boldsymbol{r}_{\bar{\mathcal{Z}}}$. Fortunately, $\boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t$ appears in the revised tableau. The update of ρ depends on the hitting times (13) and (14). These in turn depend on the numerators $-\boldsymbol{v}_i^t\boldsymbol{P}\boldsymbol{b}+\boldsymbol{v}_i^t\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}}-d_i$ and $-\boldsymbol{w}_j^t\boldsymbol{P}\boldsymbol{b}+\boldsymbol{w}_j^t\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}}-e_j$, which occur as components of the vector $\boldsymbol{U}_{\bar{\mathcal{Z}}}(-\boldsymbol{P}\boldsymbol{b}+\boldsymbol{Q}\boldsymbol{c}_{\mathcal{Z}})-\boldsymbol{c}_{\bar{\mathcal{Z}}}$, and the denominators $\boldsymbol{v}_i^t\boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}}$ and $\boldsymbol{w}_j^t\boldsymbol{P}\boldsymbol{u}_{\bar{\mathcal{Z}}}$, which occur as components of the matrix $\boldsymbol{U}_{\bar{\mathcal{Z}}}\boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t\boldsymbol{r}_{\bar{\mathcal{Z}}}$ computable from the block $\boldsymbol{U}_{\bar{\mathcal{Z}}}\boldsymbol{P}\boldsymbol{U}_{\bar{\mathcal{Z}}}^t$ of the tableau. The escape times for the active constraints also determine the update of ρ. According to Equation (16), the escape times depend on the current coefficient vector, the current value ρ_0 of ρ, and the vector $\boldsymbol{Q}^t\boldsymbol{u}_{\bar{\mathcal{Z}}} = \boldsymbol{Q}^t\boldsymbol{U}_{\bar{\mathcal{Z}}}^t\boldsymbol{r}_{\bar{\mathcal{Z}}}$, which can be computed from the block $\boldsymbol{Q}^t\boldsymbol{U}_{\bar{\mathcal{Z}}}^t$ of the tableau. Thus, the revised tableau supplies all of the ingredients for path following. Algorithm 1 outlines the steps for path following ignoring the anomalous situations.

The ingredients for handling the anomalous situations can also be read from the path tableau. The initial coefficients $\boldsymbol{r}_{\mathcal{Z}}(0) = -\boldsymbol{Q}^t\boldsymbol{u}_{\bar{\mathcal{Z}}} = -\boldsymbol{Q}^t\boldsymbol{U}_{\bar{\mathcal{Z}}}^t\boldsymbol{r}_{\bar{\mathcal{Z}}}$ are available once we sweep the tableau (17) on the diagonal entries corresponding to the constraints in $\mathcal{Z}$ at the starting point $\boldsymbol{x}(0) = -\boldsymbol{A}^{-1}\boldsymbol{b}$. As noted earlier, if the coefficients of several active constraints are simultaneously poised to exit their subdifferentials, then one must consider all possible swept and unswept combinations of these constraints. The operative criteria for choosing the right combination involve the available quantities $\boldsymbol{Q}^t\boldsymbol{u}_{\bar{\mathcal{Z}}}$ and $-\boldsymbol{Q}^t\boldsymbol{b}+\boldsymbol{R}\boldsymbol{c}_{\mathcal{Z}}$. One of the sweeping combinations is bound to give a correct direction for the next extension of the path.

The computational complexity of path following depends on the number of parameters m and the number of constraints $n = r + s$. Computation of the initial solution $-\boldsymbol{A}^{-1}\boldsymbol{b}$ takes about $3m^3$ floating point operations (flops). There is no need to store or update the $\boldsymbol{P}$ block during path following. The remaining sweeps and inverse sweeps take on the order of $n(m+n)$ flops each. This count must be multiplied by the number of segments along the path, which empirically is on the order of $O(n)$ for the small examples tried in this article. The sweep tableau requires storing $(m+n)^2$ real numbers. We recommend all computations be done in double precision. Both flop counts and storage can be halved by exploiting symmetry. Finally, it is worth mentioning some computational shortcuts for the multitask learning model. Among these are the formulas

$$\begin{aligned} (\boldsymbol{I}\otimes\boldsymbol{X})^t(\boldsymbol{I}\otimes\boldsymbol{X}) &= \boldsymbol{I}\otimes\boldsymbol{X}^t\boldsymbol{X}, \\ (\boldsymbol{I}\otimes\boldsymbol{X}^t\boldsymbol{X})^{-1} &= \boldsymbol{I}\otimes(\boldsymbol{X}^t\boldsymbol{X})^{-1}, \\ (\boldsymbol{I}\otimes\boldsymbol{X}^t\boldsymbol{X})^{-1}(\boldsymbol{I}\otimes\boldsymbol{V}) &= \boldsymbol{I}\otimes(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{V}, \\ (\boldsymbol{I}\otimes\boldsymbol{X}^t\boldsymbol{X})^{-1}(\boldsymbol{I}\otimes\boldsymbol{W}) &= \boldsymbol{I}\otimes(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{W}. \end{aligned}$$

Algorithm 1 Solution path of the primal problem (6) when $\boldsymbol{A}$ is positive definite.

Initialize $k = 0$, $\rho_0 = 0$, and the path tableau (17). Sweep the diagonal entries of $-\boldsymbol{A}$. Enter the main loop.
repeat
  Increment k by 1.
  Compute the hitting time or exit time $\rho^{(i)}$ for each constraint i.
  Set $\rho_k = \min\{\rho^{(i)} : \rho^{(i)} > \rho_{k-1}\}$.
  Update the coefficient vector by Equation (16).
  Sweep the diagonal entry of the inactive constraint that becomes active or inverse sweep the diagonal entry of the active constraint that becomes inactive.
  Update the solution vector $\boldsymbol{x}_k = \boldsymbol{x}(\rho_k)$ by Equation (11).
until $\mathcal{N}_{\mathrm{E}} = \mathcal{P}_{\mathrm{E}} = \mathcal{P}_{\mathrm{I}} = \emptyset$.

5. EXTENSIONS OF THE PATH ALGORITHM

As just presented, the path algorithm starts from the unconstrained solution and moves forward along the path to the constrained solution. With minor modifications, the same algorithm can start in the middle of the path or move in the reverse direction along it. The latter tactic proves useful in Lasso and fused-Lasso problems, where the fully constrained solution is trivial. In general, consider starting from $\boldsymbol{x}(\rho_0)$ at a point ρ_0 on the path. Let $\mathcal{Z} = \mathcal{Z}_{\mathrm{E}} \cup \mathcal{Z}_{\mathrm{I}}$ continue to denote the zero set for the segment containing ρ_0. Path following begins by sweeping the upper-left block of the tableau (18) and then proceeds as indicated in Algorithm 1. Traveling in the reverse direction entails calculation of hitting and exit times for decreasing ρ rather than increasing ρ.

Two assumptions limit the applications of Algorithm 1. The assumption that $\boldsymbol{A}$ is positive definite automatically excludes underdetermined statistical problems with more parameters than cases. The linear independence assumption on constraint vectors $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$ precludes certain regularization problems, such as the sparse fused Lasso and the two- or higher-dimensional fused Lasso. In this section, we indicate how to carry out the exact penalty method when positive definiteness of $\boldsymbol{A}$ fails and the sweep operator cannot be brought into play. Relaxation of the second restriction is more subtle and we briefly discuss the difficulties.

In the absence of constraints, $f(\boldsymbol{x})$ lacks a minimum if and only if either $\boldsymbol{A}$ has a negative eigenvalue or the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has no solution. In either circumstance, a unique global minimum may exist if enough constraints are enforced. Suppose $\boldsymbol{x}(\rho_0)$ supplies the minimum of the exact penalty function $\mathcal{E}_\rho(\boldsymbol{x})$ at $\rho = \rho_0 > 0$. Let the matrix $\boldsymbol{U}_{\mathcal{Z}}$ summarize the active constraint vectors. As we slide along the active constraints, the minimum point can be represented as $\boldsymbol{x}(\rho) = \boldsymbol{x}(\rho_0) + \boldsymbol{Y}\boldsymbol{y}(\rho)$, where the columns of $\boldsymbol{Y}$ are orthogonal to the rows of $\boldsymbol{U}_{\mathcal{Z}}$. One can construct $\boldsymbol{Y}$ by the Gram–Schmidt process; $\boldsymbol{Y}$ is then the orthogonal

complement of $\boldsymbol{U}_{\mathcal{Z}}$ in the QR decomposition. The active constraints hold in view of the identity $\boldsymbol{U}_{\mathcal{Z}}\boldsymbol{x}(\rho) = \boldsymbol{U}_{\mathcal{Z}}\boldsymbol{x}(\rho_0) = \boldsymbol{c}_{\mathcal{Z}}$.

The analog of the stationarity condition (7) under reparameterization is

$$\mathbf{0} = \boldsymbol{Y}^t\boldsymbol{A}\boldsymbol{Y}\boldsymbol{y}(\rho) + \boldsymbol{Y}^t\boldsymbol{b} + \rho\boldsymbol{Y}^t\boldsymbol{u}_{\bar{\mathcal{Z}}}. \tag{19}$$

The active constraints do not appear in this equation because $\boldsymbol{v}_i^t\boldsymbol{Y} = \mathbf{0}$ and $\boldsymbol{w}_j^t\boldsymbol{Y} = \mathbf{0}$ for i or j active. Solving for $\boldsymbol{y}(\rho)$ and $\boldsymbol{x}(\rho)$ gives

$$\begin{aligned}\boldsymbol{y}(\rho) &= -(\boldsymbol{Y}^t\boldsymbol{A}\boldsymbol{Y})^{-1}(\boldsymbol{Y}^t\boldsymbol{b} + \rho\boldsymbol{Y}^t\boldsymbol{u}_{\bar{\mathcal{Z}}}),\\ \boldsymbol{x}(\rho) &= \boldsymbol{x}(\rho_0) - \boldsymbol{Y}(\boldsymbol{Y}^t\boldsymbol{A}\boldsymbol{Y})^{-1}(\boldsymbol{Y}^t\boldsymbol{b} + \rho\boldsymbol{Y}^t\boldsymbol{u}_{\bar{\mathcal{Z}}}),\end{aligned} \tag{20}$$

and does not require inverting $\boldsymbol{A}$. Because the solution $\boldsymbol{x}(\rho)$ is affine in ρ, it is straightforward to calculate the hitting times for the inactive constraints.

Under the original parameterization, the Lagrange multipliers and corresponding active coefficients appearing in the stationarity condition (7) can still be recovered by invoking Equation (9). Again it is a simple matter to calculate exit times. The formulas are not quite as elegant as those based on the sweep operator, but all essential elements for traversing the path are available. Adding or deleting a row of the matrix $\boldsymbol{U}_{\mathcal{Z}}$ can be accomplished by updating the QR decomposition. The fast algorithms for this purpose simultaneously update $\boldsymbol{Y}$ (Lawson and Hanson 1987; Nocedal and Wright 2006). More generally, for equality-constrained problems generated by the Lasso and generalized Lasso, the constraint matrix $\boldsymbol{U}_{\mathcal{Z}}$, as one approaches the penalized solution, is often very sparse. Computation of the QR decomposition from scratch is then numerically cheap.

When the active constraint vectors are linearly dependent, $\boldsymbol{U}_{\mathcal{Z}}$ does not have full row rank. This causes problems if one determines path coefficients via Equation (9). Replacing the inverse $(\boldsymbol{U}_{\mathcal{Z}}\boldsymbol{U}_{\mathcal{Z}}^t)^{-1}$ by the Moore–Penrose pseudoinverse $(\boldsymbol{U}_{\mathcal{Z}}\boldsymbol{U}_{\mathcal{Z}}^t)^{+}$ yields the coefficient vector $\boldsymbol{r}_{\mathcal{Z}}(\rho) = (\boldsymbol{s}_{\mathcal{Z}}(\rho)^t, \boldsymbol{t}_{\mathcal{Z}}(\rho)^t)^t$ with minimal l_2 norm (Magnus and Neudecker 1999). However, exit times predicated on this version of the coefficient vector are inappropriate because, at the predicted exit time, there could exist another version of the coefficient vector $\boldsymbol{r}_{\mathcal{Z}}$ lying in the interior of the permissible range (8) with a larger l_2 norm. The set defined by the subdifferential constraints on the active coefficients is a convex polytope (a compact and polyhedral set). Its image under matrix multiplication by $\rho\boldsymbol{U}_{\mathcal{Z}}^t$ is also a convex polytope. Thus, the exit time for the active constraints is the maximum ρ going forward for which $-\boldsymbol{A}\boldsymbol{x}(\rho) - \boldsymbol{b}$ remains in the image polytope, which unfortunately is hard to determine. The dual approach taken by Tibshirani and Taylor (2011) seems somehow to circumvent the difficulty posed by naive application of the pseudoinverse solution. In practice, the whole issue can be simply resolved by computing the solution at a nearby future time $\rho + \epsilon$ using any unconstrained nonsmooth optimizer. Path following should then recommence along the direction $\boldsymbol{\beta}(\rho + \epsilon) - \boldsymbol{\beta}(\rho)$.

6. DEGREES OF FREEDOM UNDER AFFINE CONSTRAINTS

We now specialize to the least-square problem with the choices $\boldsymbol{A} = \boldsymbol{X}^t\boldsymbol{X}$, $\boldsymbol{b} = -\boldsymbol{X}^t\boldsymbol{y}$, and $\boldsymbol{x}(\rho) = \hat{\boldsymbol{\beta}}(\rho)$, and consider how to define degrees of freedom in the presence of both equality and inequality constraints. As previous authors (Efron et al. 2004; Zou, Hastie,

and Tibshirani 2007; Tibshirani and Taylor 2011) showed, the most productive approach relies on Stein's characterization (Stein 1981; Efron 2004)

$$\mathrm{df}(\hat{\boldsymbol{y}}) = \mathbf{E}\left(\sum_{i=1}^{n} \frac{\partial}{\partial y_i}\hat{y}_i\right) = \mathbf{E}\left[\mathrm{tr}(d_{\boldsymbol{y}}\hat{\boldsymbol{y}})\right]$$

of the degrees of freedom. Here, $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ is the fitted value of $\boldsymbol{y}$, and $d_{\boldsymbol{y}}\hat{\boldsymbol{y}}$ denotes its differential with respect to the entries of $\boldsymbol{y}$. Equation (11) implies that

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{XPX}^t\boldsymbol{y} + \boldsymbol{XQc}_{\mathcal{Z}} - \rho\boldsymbol{XPu}_{\bar{\mathcal{Z}}}.$$

Because ρ is fixed, it follows that $d_{\boldsymbol{y}}\hat{\boldsymbol{y}} = \boldsymbol{XPX}^t$. The representation

$$\begin{aligned}
&\boldsymbol{XPX}^t \\
&\quad = \boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t - \boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{U}_{\mathcal{Z}}^t[\boldsymbol{U}_{\mathcal{Z}}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{U}_{\mathcal{Z}}^t]^{-1}\boldsymbol{U}_{\mathcal{Z}}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t \\
&\quad = \boldsymbol{P}_1 - \boldsymbol{P}_2
\end{aligned}$$

and the cyclic permutation property of the trace function applied to the projection matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ yield the formula

$$\mathbf{E}\left[\mathrm{tr}(d_{\boldsymbol{y}}\hat{\boldsymbol{y}})\right] = m - \mathbf{E}(|\mathcal{Z}|),$$

where m equals the number of parameters. In other words, $m - |\mathcal{Z}|$ is an unbiased estimator of the degrees of freedom. This result obviously depends on our assumptions that $\boldsymbol{X}$ has full column rank m and the constraints $\boldsymbol{v}_i$ and $\boldsymbol{w}_j$ are linearly independent. The latter condition is true for Lasso and one-dimensional fused-Lasso problems. The validity of Stein's formula requires the fitted value $\hat{\boldsymbol{y}}$ to be a continuous and almost differentiable function of $\boldsymbol{y}$ for almost every $\boldsymbol{y}$ (Stein 1981). Fortunately, this is the case for Lasso (Zou, Hastie, and Tibshirani 2007) and generalized Lasso problems (Tibshirani and Taylor 2011), and for at least one case of shape-restricted regression (Meyer and Woodroofe 2000). The derivation does not depend directly on whether the constraints are equality or inequality constraints. Hence, the degrees of freedom estimator can be applied in shape-restricted regression using model selection criteria, such as C_p, AIC, and BIC, along the whole path. The concave regression example in Section 1 illustrates the general idea.

7. EXAMPLES

Our examples illustrate both the mechanics and the potential of path following. The path algorithm's ability to handle inequality constraints allows us to obtain path solutions to a variety of shape-restricted regressions. Problems of this sort may well dominate the future agenda of nonparametric estimation.

7.1 Two Toy Examples

Our first example (Lawson and Hanson 1987) fits a straight line $y = \beta_0 + x\beta_1$ to the data points (0.25,0.5), (0.5,0.6), (0.5,0.7), and (0.8,1.2) by minimizing the least-square criterion

$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ subject to the constraints

$$\beta_1 \geq 0, \quad \beta_0 \geq 0, \quad \beta_0 + \beta_1 \leq 1.$$

In our notation,

$$\boldsymbol{A} = \boldsymbol{X}^t\boldsymbol{X} = \begin{pmatrix} 4.0000 & 2.0500 \\ 2.0500 & 1.2025 \end{pmatrix}, \quad \boldsymbol{b} = -\boldsymbol{X}^t\boldsymbol{y} = \begin{pmatrix} -3.0000 \\ -1.7350 \end{pmatrix},$$

$$\boldsymbol{W} = \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \boldsymbol{e} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

The initial tableau is

$$\left(\begin{array}{rr|lllr} -4.0000 & -2.0500 & 1 & 1 & -1 & -3.0000 \\ -2.0500 & -1.2025 & 0 & 0 & -1 & -1.7350 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 \\ \hline -3.0000 & -1.7350 & 0 & 0 & -1 & 0 \end{array}\right).$$

Sweeping the first two diagonal entries produces

$$\left(\begin{array}{rr|rrr|r} 1.9794 & -3.3745 & -1.9794 & 3.3745 & -1.3951 & 0.0835 \\ -3.3745 & 6.5844 & 3.3745 & -6.5844 & 3.2099 & 1.3004 \\ \hline -1.9794 & 3.3745 & 1.9794 & -3.3745 & 1.3951 & -0.0835 \\ 3.3745 & -6.5844 & -3.3745 & 6.5844 & -3.2099 & -1.3004 \\ -1.3951 & 3.2099 & 1.3951 & -3.2099 & 1.8148 & 0.3840 \\ \hline 0.0835 & 1.3004 & -0.0835 & -1.3004 & 0.3840 & 2.5068 \end{array}\right),$$

from which we read off the unconstrained solution $\boldsymbol{\beta}(0) = (0.0835, 1.3004)^t$ and the constraint residuals $(-0.0835, -1.3004, 0.3840)^t$. The latter indicates that $\mathcal{N}_{\mathrm{I}} = \{1, 2\}$, $\mathcal{Z}_{\mathrm{I}} = \emptyset$, and $\mathcal{P}_{\mathrm{I}} = \{3\}$. Multiplying the middle block matrix by the coefficient vector $\boldsymbol{r} = (0, 0, 1)^t$ and dividing the residual vector entrywise give the hitting times $\rho = (-0.0599, 0.4051, 0.2116)$. Thus, $\rho_1 = 0.2116$ and

$$\boldsymbol{\beta}(0.2116) = \begin{pmatrix} 0.0835 \\ 1.3004 \end{pmatrix} - 0.2116 \times \begin{pmatrix} -1.3951 \\ 3.2099 \end{pmatrix} = \begin{pmatrix} 0.3787 \\ 0.6213 \end{pmatrix}.$$

Now $\mathcal{N} = \{1, 2\}$, $\mathcal{Z} = \{3\}$, $\mathcal{P} = \emptyset$, and we have found the solution. Figure 2 displays the data points and the unconstrained and constrained fitted lines.

Our second toy example concerns the toxin response problem (Schoenfeld 1986), with m toxin levels $x_1 \leq x_2 \leq \cdots \leq x_m$ and a mortality rate $y_i = f(x_i)$ at each level. It is reasonable to assume that the mortality function $f(x)$ is nonnegative and increasing. Suppose $\bar{y}_i$ are the observed death frequencies averaged across n_i trials at level x_i. In a finite sample, the $\bar{y}_i$ may fail to be nondecreasing. For example, in an Environmental Protection Agency (EPA) study of the effects of chromium on fish (Schoenfeld 1986), the observed binomial

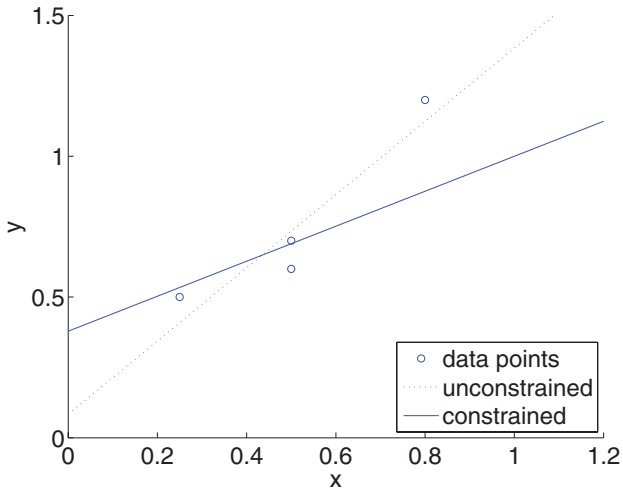

Figure 2. The data points and the fitted lines for the first toy example of constrained curve fitting (Lawson and Hanson 1987). The online version of this figure is in color.

frequencies and chromium levels are

$$\bar{\boldsymbol{y}} = (0.3752, 0.3202, 0.2775, 0.3043, 0.5327)^t,$$
$$\boldsymbol{x} = (51, 105, 194, 384, 822)^t \quad \text{in } \mu\text{g/l}.$$

Isotonic regression minimizes $\sum_{k=1}^{m}(\bar{y}_k - \theta_k)^2$ subject to the constraints $0 \le \theta_1 \le \cdots \le \theta_m$ on the binomial parameters $\theta_k = f(x_k)$. The solution path depicted in Figure 3 is continuous and piecewise linear as advertised, but the coefficient paths are nonlinear. The first four binomial parameters coalesce into the constrained estimate.

7.2 Generalized Lasso Problems

Many of the generalized Lasso problems studied by Tibshirani and Taylor (2011) reduce to minimization of some form of the objective function (6). To avoid repetition, we omit a

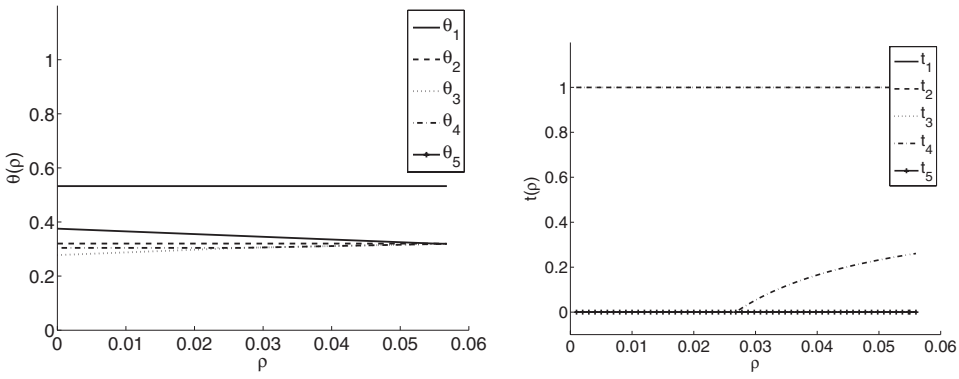

Figure 3. Toxin response example. Left: solution path. Right: coefficient paths for the constraints.

detailed discussion of this class of problems and simply refer readers interested in applications to Lasso or fused-Lasso penalized regression, outlier detections, trend filtering, and image restoration to the original article (Tibshirani and Taylor 2011). Here, we would like to point out the relevance of the generalized Lasso problems to graph-guided penalized regression (Chen et al. 2010). Suppose each node i of a graph is assigned a regression coefficient β_i and a weight w_i. In graph penalized regression, the objective function takes the form

$$\frac{1}{2}\|\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\|_2^2+\lambda_{\mathrm{G}}\sum_{i\sim j}\left|\frac{\beta_i}{\sqrt{d_i}}-\mathrm{sgn}(r_{ij})\frac{\beta_j}{\sqrt{d_j}}\right|+\lambda_{\mathrm{L}}\sum_j|\beta_j|, \tag{21}$$

where the set of neighboring pairs $i\sim j$ defines the graph, d_i is the degree of node i, and r_{ij} is the correlation coefficient between i and j. Under a line graph, the objective function (21) reduces to the fused Lasso. In two-dimensional imaging applications, the graph consists of neighboring pixels in the plane, and minimization of the function (21) is accomplished by total variation algorithms. In MRI images, the graph is defined by neighboring pixels in three dimensions. Penalties are introduced in image reconstruction and restoration to enforce smoothness. In microarray analysis, the graph reflects one or more gene networks. Smoothing the β_i over the networks is motivated by the assumption that the expression levels of related genes should rise and fall in a coordinated fashion. Ridge regularization in graph penalized regression (Li and Li 2008) is achieved by changing the objective function to

$$\frac{1}{2}\|\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\|_2^2+\lambda_{\mathrm{G}}\sum_{i\sim j}\left(\frac{\beta_i}{\sqrt{d_i}}-\mathrm{sgn}(r_{ij})\frac{\beta_j}{\sqrt{d_j}}\right)^2+\lambda_{\mathrm{L}}\sum_j|\beta_j|.$$

If one fixes either of the tuning constants in these models, our path algorithm delivers the solution path as a function of the other tuning constant. Alternatively, one can fix the ratio of the two tuning constants. Finally, the extension

$$\frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B}\|_{\mathrm{F}}^2+\lambda_{\mathrm{G}}\sum_{i\sim j}\sum_{k=1}^{K}\left|\frac{\beta_{ki}}{\sqrt{d_i}}-\mathrm{sgn}(r_{ij})\frac{\beta_{kj}}{\sqrt{d_j}}\right|+\lambda_{\mathrm{L}}\sum_{k,i}|\beta_{k,i}|$$

of the objective function to multivariate response models is obvious.

In principle, the path algorithm based on the sweep operator applies to these problems, provided the design matrix $\boldsymbol{X}$ has full column rank and the active constraints along the solution path are linearly independent. If $\boldsymbol{X}$ has reduced rank, then it is advisable to add a small amount of ridge regularization $\epsilon\sum_i\beta_i^2$ to the objective function (Tibshirani and Taylor 2011). Even so, computation of the unpenalized solution may be problematic in high dimensions. Alternatively, path following can be conducted starting from the fully constrained problem as suggested in Section 5. If the linear independence of the active constrains is violated, for example, when the graph has loops, then we recommend resorting to the numerical remedy mentioned at the end of Section 5.

7.3 Shape-Restricted Regressions

Order-constrained regression is now widely accepted as an important modeling tool (Robertson, Wright, and Dykstra 1988; Silvapulle and Sen 2005). If $\boldsymbol{\beta}$ is the parameter

vector, monotone regression includes isotone constraints $\beta_1 \le \beta_2 \le \cdots \le \beta_m$ or antitone constraints $\beta_1 \ge \beta_2 \ge \cdots \ge \beta_m$. In partially ordered regression, subsets of the parameters are subject to isotone or antitone constraints. In other problems, it is sensible to impose convex or concave constraints. If observations are collected at irregularly spaced time points $t_1 \le t_2 \le \cdots \le t_m$, then convexity translates into the constraints

$$\frac{\beta_{i+2} - \beta_{i+1}}{t_{i+2} - t_{i+1}} \ge \frac{\beta_{i+1} - \beta_i}{t_{i+1} - t_i},$$

for $1 \le i \le m - 2$. When the time intervals are uniform, these convex constraints become $\beta_{i+2} - \beta_{i+1} \ge \beta_{i+1} - \beta_i$. Concavity translates into the opposite set of inequalities. All of these shape-restricted regression problems can be solved by path following.

As an example of partial isotone regression, we fit the data from table 1.3.1 of Robertson, Wright, and Dykstra (1988) on the first-year grade point averages (GPA) of 2397 University of Iowa freshmen. These data can be downloaded as part of the R package "ic.infer." The ordinal predictors, high school rank (as a percentile) and American College Testing (ACT, a standard aptitude test) score, are discretized into nine ordered categories each. A rational admission policy based on these two predictor sets should be isotone separately within each set. Figure 4 shows the unconstrained and constrained solutions for the intercept and the two predictor sets and the solution path of the regression coefficients for the high school rank predictor.

The same authors (Robertson, Wright, and Dykstra 1988) predicted the probability of obtaining a B or better college GPA based on high school GPA and ACT score. In their data, covering 1490 college students, $\bar{y}_{ij}$ is the proportion of students who obtain a B or better college GPA among the n_{ij} students who are within the ith ACT category and the jth high school GPA category. Prediction is achieved by minimizing the criterion $\sum_i \sum_j n_{ij}(\bar{y}_{ij} - \theta_{ij})^2$ subject to the matrix partial-order constraints $\theta_{11} \ge 0$, $\theta_{ij} \le \theta_{i+1,j}$, and $\theta_{ij} \le \theta_{i,j+1}$. Figure 5 shows the solution path and the residual sum of squares and effective degrees of freedom along the path. The latter vividly illustrates the trade-off between goodness of fit and degrees of freedom. Readers can consult page 33 of Robertson, Wright, and Dykstra (1988) for the original data and the constrained parameter estimates.

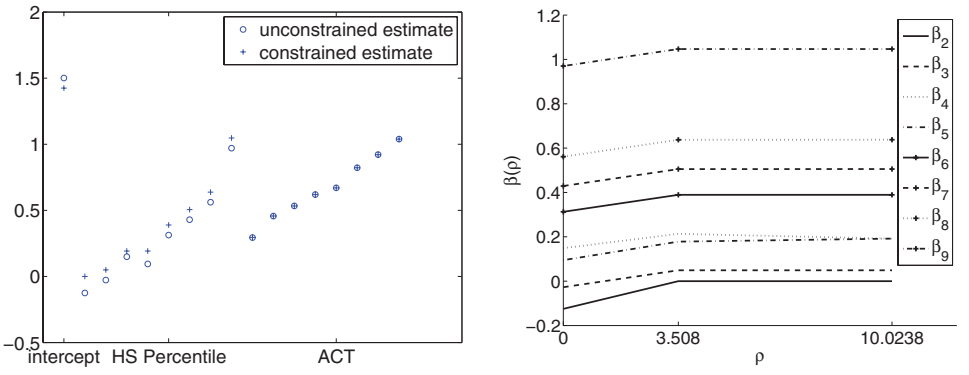

Figure 4. Left: unconstrained and constrained estimates for the Iowa GPA data. Right: solution paths of the regression coefficients corresponding to high school rank. The online version of this figure is in color.

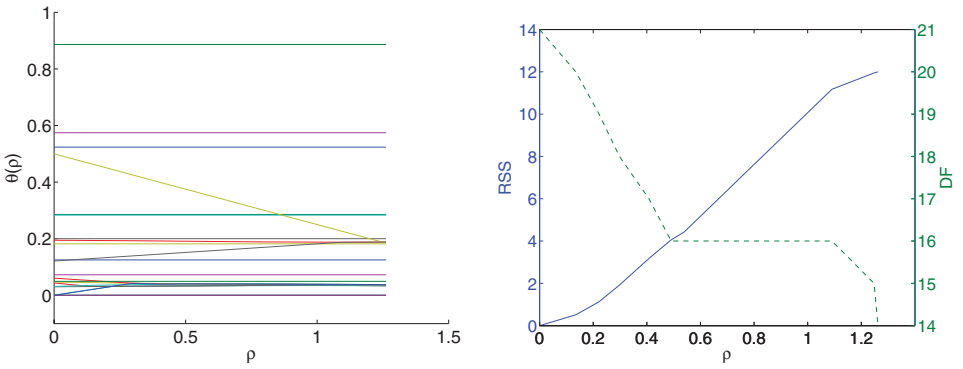

Figure 5. GPA prediction example. Left: solution path for the predicted probabilities. Right: residual sum of squares and the estimated degrees of freedom along the path. The online version of this figure is in color.

7.4 Nonparametric Shape-Restricted Regression

In this section, we visit a few problems amenable to the path algorithm arising in nonparametric statistics. Given data (x_i, y_i), $i = 1, \ldots, n$, and a weight function $w(x)$, nonparametric least squares seeks a regression function $\theta(x)$ minimizing the criterion

$$\sum_{i=1}^{n} w(x_i)[y_i - \theta(x_i)]^2 \tag{22}$$

over a space $\mathcal{C}$ of functions with shape restrictions. In concave regression, for instance, $\mathcal{C}$ is the space of concave functions. This seemingly intractable infinite-dimensional problem can be simplified by minimizing the least-square criterion (3) subject to inequality constraints. For a univariate predictor and concave regression, the constraints (4) are pertinent. The piecewise linear function extrapolated from the estimated θ_i is clearly concave. The consistency of concavity-constrained least squares is proved by Hanson and Pledger (1976); the asymptotic distribution of the corresponding estimator and its rate of convergence are investigated in later articles (Mammen 1991; Groeneboom, Jongbloed, and Wellner 2001). Other relevant shape restrictions for univariate predictors include monotonicity (Brunk 1955; Grenander 1956), convexity (Groeneboom, Jongbloed, and Wellner 2001), supermodularity (Beresteanu 2004), and combinations of these.

Multidimensional nonparametric estimation is much harder because there is no natural order on $\mathbb{R}^d$ when $d > 1$. One fruitful approach to shape-restricted regression relies on sieve estimators (Shen and Wong 1994; Beresteanu 2004). The general idea is to introduce a basis of local functions (e.g., normalized B-splines) centered on the points of a grid $\boldsymbol{G}$ spanning the support of the covariate vectors $\boldsymbol{x}_i$. Admissible estimators are then limited to linear combinations of the basis functions subject to restrictions on the estimates at the grid points. Estimation can be formalized as minimization of the criterion $\|\boldsymbol{y} - \boldsymbol{\Psi}(\boldsymbol{X})\boldsymbol{\theta}\|_2^2$ subject to the constraints $\boldsymbol{C}\boldsymbol{\Psi}(\boldsymbol{G})\boldsymbol{\theta} \preceq \boldsymbol{0}$, where $\boldsymbol{\Psi}(\boldsymbol{X})$ is the matrix of basis functions evaluated at the covariate vectors $\boldsymbol{x}_i$, $\boldsymbol{\Psi}(\boldsymbol{G})$ is the matrix of basis functions evaluated at the grid points, and $\boldsymbol{\theta}$ is a vector of regression coefficients. The linear inequality constraints incorporated in the matrix $\boldsymbol{C}$ reflect the required shape restrictions. Estimation is performed on a sequence of grids (a sieve). Controlling the rate at which the sieve sequence converges yields a consistent

estimator (Shen and Wong 1994; Beresteanu 2004). Prediction reduces to interpolation, and the path algorithm provides a computational engine for sieve estimation.

A related but different approach for multivariate convex regression minimizes the least-square criterion (3) subject to the constraints $\boldsymbol{\xi}_i^t(\boldsymbol{x}_j - \boldsymbol{x}_i) \leq \theta_j - \theta_i$ for every ordered pair (i, j). In effect, θ_i is viewed as the value of the regression function $\theta(\boldsymbol{x})$ at the point $\boldsymbol{x}_i$. The unknown vector $\boldsymbol{\xi}_i$ serves as a subgradient of $\theta(\boldsymbol{x})$ at $\boldsymbol{x}_i$. Because convexity is preserved by maxima, the formula

$$\theta(\boldsymbol{x}) = \max_j \left[\theta_j + \boldsymbol{\xi}_j^t(\boldsymbol{x} - \boldsymbol{x}_j)\right]$$

defines a convex function with value θ_i at $\boldsymbol{x} = \boldsymbol{x}_i$. In concave regression, the opposite constraint inequalities are imposed. Interpolation of predicted values in this model is accomplished by simply taking minima or maxima. Estimation reduces to a positive semidefinite quadratic program involving $n(d + 1)$ variables and $n(n - 1)$ inequality constraints. Note that the feasible region is nontrivial because setting all $\theta_i = 0$ and all $\boldsymbol{\xi}_i = \boldsymbol{0}$ works. In implementing the extension of the path algorithm mentioned in Section 5, the large number of constraints may prove to be a hindrance and lead to very short path segments. To improve estimation of the subgradients, it might be worth adding a small multiple of the ridge penalty $\sum_i \|\boldsymbol{\xi}_i\|_2^2$ to the objective function (3). This would have the beneficial effect of turning a semidefinite quadratic program into a positive definite quadratic program.

8. CONCLUSIONS

Our new path algorithm for convex quadratic programming under affine constraints generalizes previous path algorithms for Lasso penalized regression and its extensions. Our path algorithm directly attacks the primal problem; the complementary method of Tibshirani and Taylor (2011) solves the dual problem. Our various examples confirm the primal algorithm's versatility. Its potential disadvantages involve computing the initial point $-\boldsymbol{A}^{-1}\boldsymbol{b}$ and storing the sweeping tableau. In problems with large numbers of parameters, neither of these steps is trivial. However, if $\boldsymbol{A}$ has enough structure, then an explicit inverse may exist. As we have already noted, once $\boldsymbol{A}^{-1}$ is computed, there is no need to store the entire tableau. The multitask regression problem with a large number of responses per case is a typical example where computation of $\boldsymbol{A}^{-1}$ simplifies. In settings where the matrix $\boldsymbol{A}$ is singular, parameter constraints may compensate. We have briefly indicated how to conduct path following in this circumstance. Although our more stringent assumption of linear independence of the constraint gradients excludes some interesting examples treated by Tibshirani and Taylor (2011), many practical problems can be finessed by the remedy discussed in Section 5.

Our path algorithm qualifies as a general convex quadratic program solver. Custom algorithms have been developed for many special cases of quadratic programming. For example, the pool-adjacent-violators algorithm (PAVA) is now the standard approach to isotone regression (de Leeuw, Hornik, and Mair 2009). The other generic methods of quadratic programming include active set and interior point methods. For applications where only the constrained estimate is of interest, it would be hard to beat these well-honed algorithms. In regularized statistical estimation and inverse problems, the primary goal is to

select relevant predictors rather than to find a constrained solution. Thus, the entire solution path commands more interest than any single point along it, and the path algorithm's ability to deliver the whole regularized path with little additional computation cost beyond constrained estimation is bound to be appealing to statisticians. Numerical comparisons with competing methods would be illuminating but would also depend heavily on programming details and problem choices. In the interests of brevity, we refrain from making numerical comparisons here.

The path algorithm bears a stronger resemblance to the active set method (Nocedal and Wright 2006). Indeed, both operate by deleting and adding constraints to a working active set. However, they differ in at least two respects. First, the initial active set is constructed arbitrarily in the active set method. Distinct initial active sets produce different iteration sequences. In contrast, the path algorithm always starts from the unconstrained solution. The initial active set is determined as a by-product. Second, the mechanics of adding or deleting constraints differ in the two methods. The active set method chooses the direction of movement that tends to decrease the quadratic objective function most, while the path algorithm tracks the tuning constant ρ. In fact, path following steadily increases the objective function until it reaches its constrained solution. In this sense, the active set method is greedier than the path algorithm, which expends its effort in traversing the solution path.

SUPPLEMENTARY MATERIALS

MATLAB code: Data and MATLAB code for all examples in this article are available in the supplementary materials (path_quadratic.zip). The readme.txt file describes the contents of each file in the package. They are also part of the SparseReg toolbox maintained and distributed on the first author's website.

ACKNOWLEDGMENTS

We thank the editor, associate editor, and two referees, whose comments greatly improved the article. We also acknowledge support from grants GM53275, MH59490, CA87949, CA16042, R01HG006139, and NCSU FRPD.

[Received April 2011. Revised February 2012.]

REFERENCES

Beresteanu, A. (2004), "Nonparametric Estimation of Regression Functions Under Restrictions on Partial Derivatives," Working Papers 04-06, Duke University, Department of Economics. [279]

Brunk, H. D. (1955), "Maximum Likelihood Estimates of Monotone Parameters," *Annals of Mathematical Statistics*, 26, 607–616. [279]

Chen, X., Lin, Q., Kim, S., Carbonell, J., and Xing, E. (2012), "Smoothing Proximal Gradient Method for General Structured Sparse Regression," *Annals of Applied Statistics*, 6, 719–752. [269,277]

de Leeuw, J., Hornik, K., and Mair, P. (2009), "Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods," *Journal of Statistical Software*, 32 (5), 1–24. [280]

Dempster, A. P. (1969), *Elements of Continuous Multivariate Analysis* (Addison-Wesley Series in Behavioral Sciences), Reading, MA: Addison-Wesley. [269]

Efron, B. (2004), "The Estimation of Prediction Error: Covariance Penalties and Cross-Validation" (with discussion), *Journal of the American Statistical Association*, 99, 619–642. [263,274]

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression" (with discussion), *The Annals of Statistics*, 32, 407–499. [262,273]

Friedman, J. (2008), "Fast Sparse Regression and Classification," [online] in *Proceedings of the 23rd International Workshop on Statistical Modelling*, pp. 27–57. Available at *http://www-stat.stanford.edu/~jhf/ftp/GPSpaper.pdf* [262]

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007), "Pathwise Coordinate Optimization," *Annals of Applied Statistics*, 1, 302–332. [269]

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *The American Statistician*, 33, 149–158. [269]

Grenander, U. (1956), "On the Theory of Mortality Measurement. Part II," *Skand Aktuarietidskr*, 39, 125–153. [279]

Groeneboom, P., Jongbloed, G., and Wellner, J. A. (2001), "Estimation of a Convex Function: Characterizations and Asymptotic Theory," *The Annals of Statistics*, 29, 1653–1698. [263,279]

Hanson, D. L., and Pledger, G. (1976), "Consistency in Concave Regression," *The Annals of Statistics*, 4, 1038–1050. [263,279]

Hildreth, C. (1954), "Point Estimates of Ordinates of Concave Functions," *Journal of the American Statistical Association*, 49, 598–619. [263]

Jennrich, R. (1977), "Stepwise Regression," in *Statistical Methods for Digital Computers*, eds. A. Ralston, K. Enslein, and H. S. Wilf, New York: Wiley-Interscience, pp. 58–75. [269]

Lange, K. (2010), *Numerical Analysis for Statisticians* (2nd ed., Statistics and Computing), New York: Springer. [269]

Lawson, C. L., and Hanson, R. J. (1987), *Solving Least Squares Problems* (New ed., Classics in Applied Mathematics), Philadelphia, PA: Society for Industrial and Applied Mathematics. [273,274]

Li, C., and Li, H. (2008), "Network-Constrained Regularization and Variable Selection for Analysis of Genomic Data," *Bioinformatics*, 24, 1175–1182. [277]

Little, R. J. A., and Rubin, D. B. (2002), *Statistical Analysis With Missing Data* (2nd ed., Wiley Series in Probability and Statistics), Hoboken, NJ: Wiley-Interscience. [269]

Liu, J., Yuan, L., and Ye, J. (2010), "An Efficient Algorithm for a Class of Fused Lasso Problems," in *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 323–332. [269]

Magnus, J. R., and Neudecker, H. (1999), *Matrix Differential Calculus With Applications in Statistics and Econometrics* (Wiley Series in Probability and Statistics), Chichester: Wiley. [273]

Mammen, E. (1991), "Nonparametric Regression Under Qualitative Smoothness Assumptions," *The Annals of Statistics*, 19, 741–759. [263,279]

Meyer, M., and Woodroofe, M. (2000), "On the Degrees of Freedom in Shape-Restricted Regression," *The Annals of Statistics*, 28, 1083–1104. [274]

Nocedal, J., and Wright, S. J. (2006), *Numerical Optimization* (2nd ed., Springer Series in Operations Research and Financial Engineering), New York: Springer. [273,281]

Robertson, T., Wright, F. T., and Dykstra, R. L. (1988), *Order Restricted Statistical Inference* (Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics), Chichester: Wiley. [277,278]

Rosset, S., and Zhu, J. (2007), "Piecewise Linear Regularized Solution Paths," *The Annals of Statistics*, 35, 1012–1030. [262]

Ruszczyński, A. (2006), *Nonlinear Optimization*, Princeton, NJ: Princeton University Press. [265]

Savage, C. (1997), "A Survey of Combinatorial Gray Codes," *SIAM Review*, 39, 605–629. [269]

Schoenfeld, D. A. (1986), "Confidence Bounds for Normal Means Under Order Restrictions, With Application to Dose-Response Curves, Toxicology Experiments, and Low-Dose Extrapolation," *Journal of the American Statistical Association*, 81, 186–195. [275]

Shen, X., and Wong, W. H. (1994), "Convergence Rate of Sieve Estimates," *The Annals of Statistics*, 22, 580–615. [279]

Silvapulle, M. J., and Sen, P. K. (2005), *Constrained Statistical Inference: Inequality, Order, and Shape Restrictions* (Wiley Series in Probability and Statistics), Hoboken, NJ: Wiley-Interscience. [277]

Stein, C. M. (1981), "Estimation of the Mean of a Multivariate Normal Distribution," *The Annals of Statistics*, 9, 1135–1151. [274]

Tibshirani, R., and Taylor, J. (2011), "The Solution Path of the Generalized Lasso," *The Annals of Statistics*, 39, 1335–1371. [262,265,273,274,276,277,280]

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. (2011), "Nearly-Isotonic Regression," *Technometrics*, 53, 54–61. [264]

Wu, T. T., and Lange, K. (2008), "Coordinate Descent Algorithms for Lasso Penalized Regression," *Annals of Applied Statistics*, 2, 224–244. [269]

Zou, H., Hastie, T., and Tibshirani, R. (2007), "On the 'Degrees of Freedom' of the Lasso," *The Annals of Statistics*, 35, 2173–2192. [262,274]