Specifically for the Lasso, one alternative strategy for logistic regression is to use a quadratic approximation for the log-likelihood. Consider the Bayesian version of Lasso with hyperparameter γ (i.e., the penalized rather than constrained version of Lasso):

$$\begin{aligned}
&\log f(\boldsymbol{\beta}|y_1,\ldots,y_n)\\
&\quad\propto \sum_{i=1}^{n}\log\bigl(y_i\Lambda(\mathbf{x}_i\boldsymbol{\beta})+(1-y_i)\bigl(1-\Lambda(\mathbf{x}_i\boldsymbol{\beta})\bigr)\bigr)+d\log\biggl(\frac{\gamma^{1/2}}{2}\biggr)-\gamma^{1/2}\sum_{i=1}^{d}|\beta_i|\\
&\quad\approx \biggl(\sum_{i=1}^{n}a_i(\mathbf{x}_i\boldsymbol{\beta})^2+b_i(\mathbf{x}_i\boldsymbol{\beta})+c_i\biggr)+d\log\biggl(\frac{\gamma^{1/2}}{2}\biggr)-\gamma^{1/2}\sum_{i=1}^{d}|\beta_i|,
\end{aligned}$$

where Λ denotes the logistic link, d is the dimension of $\boldsymbol{\beta}$ and a_i, b_i and c_i are Taylor coefficients. Fu's elegant coordinatewise "Shooting algorithm" [Fu (1998)], can optimize this target starting from either the least squares solution or from zero. In our experience the shooting algorithm converges rapidly.

REFERENCES

BREIMAN, L. (2001). Random forests. Available at ftp://ftp.stat.berkeley.edu/pub/users/breiman/randomforest2001.pdf.

FU, W. J. (1998). Penalized regressions: The Bridge versus the Lasso. *J. Comput. Graph. Statist.* **7** 397–416.

OSBORNE, M. R., PRESNELL, B. and TURLACH, B. A. (2000). A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.* **20** 389–403.

RIDGEWAY, G. (2003). GBM 0.7-2 package manual. Available at http://cran.r-project.org/doc/packages/gbm.pdf.

AVAYA LABS RESEARCH
AND
DEPARTMENT OF STATISTICS
RUTGERS UNIVERSITY
PISCATAWAY, NEW JERSEY 08855
USA
E-MAIL: madigan@stat.rutgers.edu

RAND STATISTICS GROUP
SANTA MONICA, CALIFORNIA 90407-2138
USA
E-MAIL: gregr@rand.org

DISCUSSION

BY SAHARON ROSSET AND JI ZHU

IBM T. J. Watson Research Center and Stanford University

1. Introduction. We congratulate the authors on their excellent work. The paper combines elegant theory and useful practical results in an intriguing manner. The LAR–Lasso–boosting relationship opens the door for new insights on existing

methods' underlying statistical mechanisms and for the development of new and promising methodology. Two issues in particular have captured our attention, as their implications go beyond the squared error loss case presented in this paper, into wider statistical domains: robust fitting, classification, machine learning and more. We concentrate our discussion on these two results and their extensions.

2. Piecewise linear regularized solution paths. The first issue is the piecewise linear solution paths to regularized optimization problems. As the discussion paper shows, the path of optimal solutions to the "Lasso" regularized optimization problem

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{1}$$

is piecewise linear as a function of λ; that is, there exist $\infty > \lambda_0 > \lambda_1 > \cdots > \lambda_m = 0$ such that $\forall\, \lambda \geq 0$, with $\lambda_k \geq \lambda \geq \lambda_{k+1}$, we have

$$\hat{\beta}(\lambda) = \hat{\beta}(\lambda_k) - (\lambda - \lambda_k)\gamma_k.$$

In the discussion paper's terms, γ_k is the "LAR" direction for the kth step of the LAR–Lasso algorithm.

This property allows the LAR–Lasso algorithm to generate the whole path of Lasso solutions, $\hat{\beta}(\lambda)$, for "practically" the cost of one least squares calculation on the data (this is exactly the case for LAR but not for LAR–Lasso, which may be significantly more computationally intensive on some data sets). The important practical consequence is that it is not necessary to select the regularization parameter λ in advance, and it is now computationally feasible to optimize it based on cross-validation (or approximate C_p, as presented in the discussion paper).

The question we ask is: what makes the solution paths piecewise linear? Is it the use of squared error loss? Or the Lasso penalty? The answer is that both play an important role. However, the family of (loss, penalty) pairs which facilitates piecewise linear solution paths turns out to contain many other interesting and useful optimization problems.

We now briefly review our results, presented in detail in Rosset and Zhu (2004). Consider the general regularized optimization problem

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \sum_i L(y_i, \mathbf{x}_i^t\beta) + \lambda J(\beta), \tag{2}$$

where we only assume that the loss L and the penalty J are both convex functions of β for any sample $\{\mathbf{x}_i^t, y_i\}_{i=1}^n$. For our discussion, the data sample is assumed fixed, and so we will use the notation $L(\beta)$, where the dependence on the data is implicit.

Notice that piecewise linearity is equivalent to requiring that

$$\frac{\partial\hat{\beta}(\lambda)}{\partial\lambda} \in \mathcal{R}^p$$

is piecewise constant as a function of λ. If L, J are twice differentiable functions of β, then it is easy to derive that

$$\frac{\partial \hat{\beta}(\lambda)}{\partial \lambda} = -\big(\nabla^2 L(\hat{\beta}(\lambda)) + \lambda \nabla^2 J(\hat{\beta}(\lambda))\big)^{-1} \nabla J\big(\hat{\beta}(\lambda)\big). \tag{3}$$

With a little more work we extend this result to "almost twice differentiable" loss and penalty functions (i.e., twice differentiable everywhere except at a finite number of points), which leads us to the following *sufficient conditions for piecewise linear* $\hat{\beta}(\lambda)$:

1. $\nabla^2 L(\hat{\beta}(\lambda))$ is piecewise constant as a function of λ. This condition is met if L is a piecewise-quadratic function of β. This class includes the squared error loss of the Lasso, but also absolute loss and combinations of the two, such as Huber's loss.
2. $\nabla J(\hat{\beta}(\lambda))$ is piecewise constant as a function of λ. This condition is met if J is a piecewise-linear function of β. This class includes the l_1 penalty of the Lasso, but also the l_∞ norm penalty.

2.1. *Examples.* Our first example is the "Huberized" Lasso; that is, we use the loss

$$L(y, \mathbf{x}\beta) = \begin{cases} (y - \mathbf{x}^t\beta)^2, & \text{if } |y - \mathbf{x}^t\beta| \le \delta, \\ \delta^2 + 2\delta(|y - \mathbf{x}\beta| - \delta), & \text{otherwise,} \end{cases} \tag{4}$$

with the Lasso penalty. This loss is more robust than squared error loss against outliers and long-tailed residual distributions, while still allowing us to calculate the whole path of regularized solutions efficiently.

To illustrate the importance of robustness in addition to regularization, consider the following simple simulated example: take $n = 100$ observations and $p = 80$ predictors, where all x_{ij} are i.i.d. $N(0, 1)$ and the true model is

$$y_i = 10 \cdot x_{i1} + \varepsilon_i, \tag{5}$$

$$\varepsilon_i \overset{\text{i.i.d.}}{\sim} 0.9 \cdot N(0, 1) + 0.1 \cdot N(0, 100). \tag{6}$$

So the normality of residuals, implicitly assumed by using squared error loss, is violated.

Figure 1 shows the optimal coefficient paths $\hat{\beta}(\lambda)$ for the Lasso (right) and "Huberized" Lasso, using $\delta = 1$ (left). We observe that the Lasso fails in identifying the correct model $E(Y|x) = 10x_1$ while the robust loss identifies it almost exactly, *if we choose the appropriate regularization parameter*.

As a second example, consider a classification scenario where the loss we use depends on the margin $y\mathbf{x}^t\beta$ rather than on the residual. In particular, consider the 1-*norm* support vector machine regularized optimization problem, popular in the

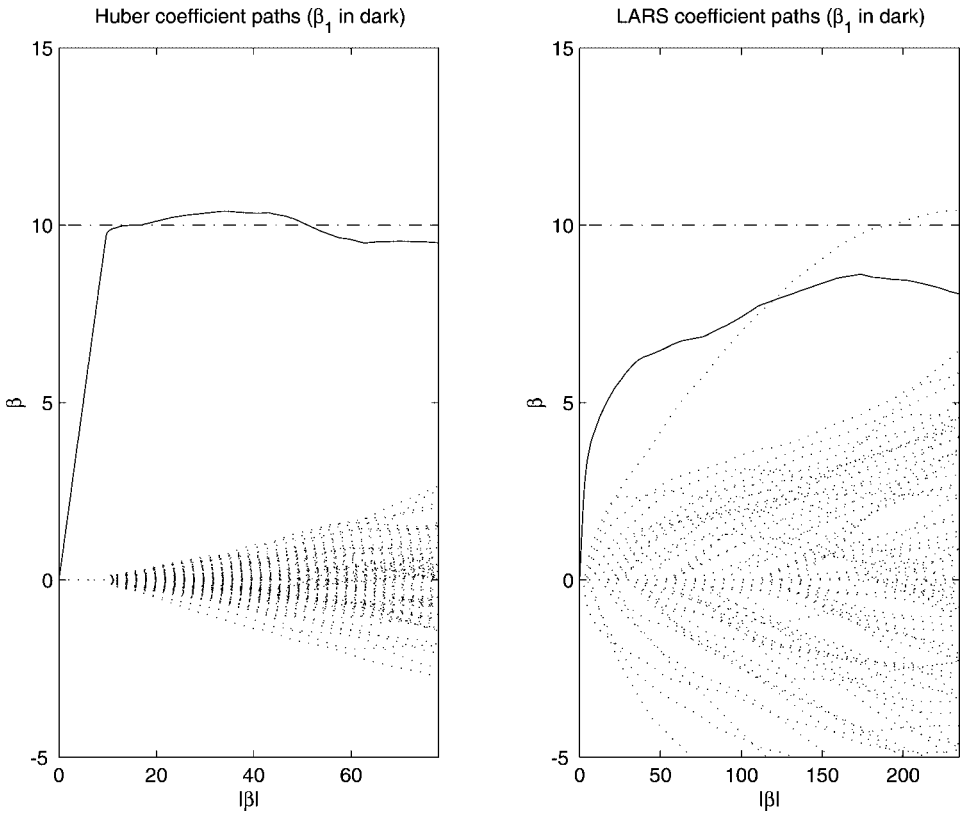

FIG. 1. *Coefficient paths for Huberized Lasso* (left) *and Lasso* (right) *for data example*: $\hat{\beta}_1(\lambda)$ *is the full line; the true model is* $E(Y|x) = 10x_1$.

machine learning community. It consists of minimizing the "hinge loss" with a Lasso penalty:

$$L(y, \mathbf{x}^t\beta) = \begin{cases} (1 - y\mathbf{x}^t\beta), & \text{if } y\mathbf{x}^t\beta \le 1, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This problem obeys our conditions for piecewise linearity, and so we can generate all regularized solutions for this fitting problem efficiently. This is particularly advantageous in high-dimensional machine learning problems, where regularization is critical, and it is usually not clear in advance what a good regularization parameter would be. A detailed discussion of the computational and methodological aspects of this example appears in Zhu, Rosset, Hastie, and Tibshirani (2004).

3. Relationship between "boosting" algorithms and l_1-regularized fitting. The discussion paper establishes the close relationship between ε-stagewise linear regression and the Lasso. Figure 1 in that paper illustrates the near-equivalence in

the solution paths generated by the two methods, and Theorem 2 formally states a related result. It should be noted, however, that their theorem falls short of proving the global relation between the methods, which the examples suggest.

In Rosset, Zhu and Hastie (2003) we demonstrate that this relation between the path of l_1-regularized optimal solutions [which we have denoted above by $\hat{\beta}(\lambda)$] and the path of "generalized" ε-stagewise (AKA boosting) solutions extends beyond squared error loss and in fact applies to any convex differentiable loss.

More concretely, consider the following generic gradient-based "ε-boosting" algorithm [we follow Friedman (2001) and Mason, Baxter, Bartlett and Frean (2000) in this view of boosting], which iteratively builds the solution path $\beta^{(t)}$:

ALGORITHM 1 (Generic gradient-based boosting algorithm).

1. Set $\beta^{(0)} = 0$.
2. For $t = 1 : T$,
   (a) Let $j_t = \arg\max_j |\frac{\partial \sum_i L(y_i, \mathbf{x}_i^t \beta^{(t-1)})}{\partial \beta_j^{(t-1)}}|$.
   (b) Set $\beta_{j_t}^{(t)} = \beta_{j_t}^{(t-1)} - \varepsilon \operatorname{sign}(\frac{\partial \sum_i L(y_i, \mathbf{x}_i^t \beta^{(t-1)})}{\partial \beta_{j_t}^{(t-1)}})$ and $\beta_k^{(t)} = \beta_k^{(t-1)}$, $k \neq j_t$.

This is a coordinate descent algorithm, which reduces to forward stagewise, as defined in the discussion paper, if we take the loss to be squared error loss: $L(y_i, \mathbf{x}_i^t \beta^{(t-1)}) = (y_i - \mathbf{x}_i^t \beta^{(t-1)})^2$. If we take the loss to be the exponential loss,

$$L\big(y_i, \mathbf{x}_i^t \beta^{(t-1)}\big) = \exp\big(-y_i \mathbf{x}_i^t \beta^{(t-1)}\big),$$

we get a variant of AdaBoost [Freund and Schapire (1997)]—the original and most famous boosting algorithm.

Figure 2 illustrates the equivalence between Algorithm 1 and the optimal solution path for a simple logistic regression example, using five variables from the "spam" dataset. We can see that there is a perfect equivalence between the regularized solution path (left) and the "boosting" solution path (right).

In Rosset, Zhu and Hastie (2003) we formally state this equivalence, with the required conditions, as a conjecture. We also generalize the weaker result, proven by the discussion paper for the case of squared error loss, to any convex differentiable loss.

This result is interesting in the boosting context because it facilitates a view of boosting as approximate and implicit regularized optimization. The situations in which boosting is employed in practice are ones where explicitly solving regularized optimization problems is not practical (usually very high-dimensional predictor spaces). The approximate regularized optimization view which emerges from our results allows us to better understand boosting and its great empirical success [Breiman (1999)]. It also allows us to derive approximate convergence results for boosting.

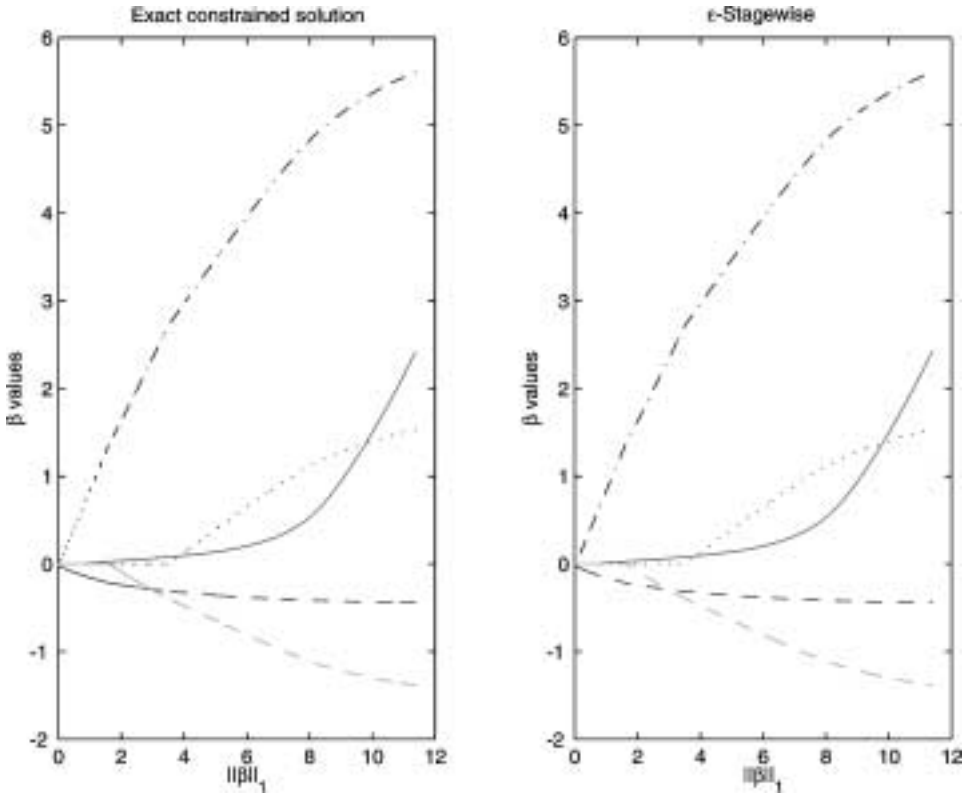

FIG. 2. *Exact coefficient paths* (left) *for* l_1*-constrained logistic regression and boosting coefficient paths* (right) *with binomial log-likelihood loss on five variables from the "spam" dataset. The boosting path was generated using* $\varepsilon = 0.003$ *and* 7000 *iterations.*

4. Conclusion. The computational and theoretical results of the discussion paper shed new light on variable selection and regularization methods for linear regression. However, we believe that variants of these results are useful and applicable beyond that domain. We hope that the two extensions that we have presented convey this message successfully.

Acknowledgment. We thank Giles Hooker for useful comments.

REFERENCES

BREIMAN, L. (1999). Prediction games and arcing algorithms. *Neural Computation* **11** 1493–1517.

FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139.

FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232.

MASON, L., BAXTER, J., BARTLETT, P. and FREAN, M. (2000). Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems* **12** 512–518. MIT Press, Cambridge, MA.

ROSSET, S. and ZHU, J. (2004). Piecewise linear regularized solution paths. *Advances in Neural Information Processing Systems* **16**. To appear.

ROSSET, S., ZHU, J. and HASTIE, T. (2003). Boosting as a regularized path to a maximum margin classifier. Technical report, Dept. Statistics, Stanford Univ.

ZHU, J., ROSSET, S., HASTIE, T. and TIBSHIRANI, R. (2004). 1-norm support vector machines. *Neural Information Processing Systems* **16**. To appear.

IBM T. J. WATSON RESEARCH CENTER
P.O. BOX 218
YORKTOWN HEIGHTS, NEW YORK 10598
USA
E-MAIL: srosset@us.ibm.com

DEPARTMENT OF STATISTICS
UNIVERSITY OF MICHIGAN
550 EAST UNIVERSITY
ANN ARBOR, MICHIGAN 48109-1092
USA
E-MAIL: jizhu@umich.edu

DISCUSSION

BY ROBERT A. STINE

University of Pennsylvania

I have enjoyed reading the work of each of these authors over the years, so it is a real pleasure to have this opportunity to contribute to the discussion of this collaboration. The geometry of LARS furnishes an elegant bridge between the Lasso and Stagewise regression, methods that I would not have suspected to be so related. Toward my own interests, LARS offers a rather different way to construct a regression model by gradually blending predictors rather than using a predictor all at once. I feel that the problem of "automatic feature generation" (proposing predictors to consider in a model) is a current challenge in building regression models that can compete with those from computer science, and LARS suggests a new approach to this task. In the examples of Efron, Hastie, Johnstone and Tibshirani (EHJT) (particularly that summarized in their Figure 5), LARS produces models with smaller predictive error than the old workhorse, stepwise regression. Furthermore, as an added bonus, the code supplied by the authors runs faster for me than the `step` routine for stepwise regression supplied with *R*, the generic version of S-PLUS that I use.

My discussion focuses on the use of C_p to choose the number of predictors. The bootstrap simulations in EHJT show that LARS reaches higher levels of "proportion explained" than stepwise regression. Furthermore, the goodness-of-fit obtained by LARS remains high over a wide range of models, in sharp contrast to the narrow peak produced by stepwise selection. Because the cost of overfitting with LARS appears less severe than with stepwise, LARS would seem to have a clear advantage in this respect. Even if we do overfit, the fit of LARS degrades