Response to Associate Editor

February 10, 2017

We appreciate the helpful feedback from the reviewer. We have addressed your questions and comments. Below we give a point-by-point response to each of the questions:

1. The authors provided insufficient justification for using a large number of regularization parameters

   We have updated the introduction with more examples of problems with multiple regularization parameters. We inserted the following paragraph into Section 1:

   > In recent years, there has been much interest in combining regularization methods to produce models with multiple desired characteristics. For example, the elastic net combines the lasso and ridge penalties; and the sparse group lasso combines the group lasso and lasso penalties. In Bayesian regression, a popular method for pruning irrelevant features is to use automatic relevance determination, which associates each feature with a separate regularization parameter. Finally, neural networks commonly use regularization to control the weights at each node. showed that using separate regularization parameters for each layer in a neural network can actually improve performance. From a theoretical viewpoint, certain regression problems require multiple regularization parameters to achieve oracle convergence rates, such as additive models with different smoothness.

2. Some important details have been omitted from the empirical results. Full reproducibility is expected

   We apologize for omitting some simulation details. We included the number of simulation runs used in Section 3. We also specify the parameters used in the gradient descent procedure in Section 2.5.

3. The empirical results cover a relatively small range of scenarios

   Thank you for the helpful feedback. We added a new example of matrix completion to illustrate the wide applicability of our method. This example moves away from the simple regression framework and considers matrix-valued data with partially observed entries. The problem now involves minimizing a penalized loss with a nuclear norm penalty. This joint optimization problem has a much more complex differentiable space compared to the other examples. We had to rely on different representations of this differentiable space in order to (1) prove that the conditions of Theorem 1 were satisfied and (2) calculate the gradient.

   The new sections are as follows. Section 2.4.4 introduces low-rank matrix completion and illustrates how to transform the joint optimization problem into an equivalent smooth joint optimization problem. Section 3.4 provides simulation results. Section 1.3.4 in the Appendix provides more details on how to calculate the gradient and shows the conditions in Theorem 1 are satisfied.

4. The technical conditions seem quite restrictive from a practical point of view, and need further explanation/justification (or weakening)

   We apologize for the confusion regarding the technical conditions. Reviewer 2 was concerned that our paper would not be applicable to high-dimensional problems since we had previously specified that the objective function must be strictly convex. Reviewer 2 is correct that our paper does not actually need the strict convexity assumption. We have removed this from the text. Our results only depend on the three conditions in Theorem 1. These conditions include many popular penalized regression settings.

5. Make sure to provide all code for all experiments

   We have included all the code for our experiments. In addition, we plan to make our code fully available on Github.

6. They state in the abstract and in the paper: ”For many penalized regression problems, the validation loss is actually smooth almost-everywhere with respect to the penalty parameters.” I assume that almost everywhere means ”almost everywhere with respect to Lebesgue measure.” But of course, this same statement is true of the objective itself, for which gradient descent cannot be used. The relevant condition seems to be to be whether the loss is smooth almost everywhere with respect to the probability measure induced by the true sampling model, which is not the case for e.g. lasso/group lasso/etc. Can the authors please clarify and elaborate on this point?

   We agree that the condition of interest is whether the loss is smooth almost everywhere with respect to the probability measure induced by the true sampling model. We suspect that the set of knots, the penalty parameters at which the validation loss is not differentiable, has measure zero under the true sampling model. Gradient descent is agnostic to whether or not the training criterion is smooth, so it seems unlikely that it will prefer knots.

   We also investigated whether the minimizer of the validation loss tended to be at knots. We performed a simulation study with a penalized least squares problem with a lasso penalty. The penalty parameter that minimized the validation loss was never located at a knot. This simple simulation study suggests that it is likely that in general, the penalty parameter that minimizes the validation loss is unlikely to be a knot.

   Simulation settings We considered a linear model with 50 covariates. The training and validation sets included 15 and 10 observations, respectively. The response was generated data from the model

   $$y = X\beta + \sigma\epsilon$$

   where $\beta = (1, 1, 1, 0, ..., 0)$. ϵ and X were generated from a standard Gaussian distribution. We fit models that minimized the penalized training criterion

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

   To find the lasso parameter that minimized the validation loss, we tested all the points along the lasso path as well as 2000 points in between each pair of consecutive knots.