Response to Associate Editor

December 14, 2016

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 posed by the reviewer:

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

   We have updated Section 1 with more examples of problems with multiple regularization parameters.

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

   We have included more details in Section 3 regarding the simulation studies.

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

   Adding a new experiment?

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

   As noted by Reviewer 2, our paper does not need the strict convexity assumption that we have mistakenly included. This assumption has been removed. Our technical conditions as written are now applicable to many popular penalized regression settings.

5. Make sure to provide all code for all experiments

   It is.

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?

   Algorithmically, at each iteration of gradient descent, it is unlikely for us to end up at a penalty parameter where the valiation loss is not smooth. This can be viewed as analogous to iteratively solving the lasso using subgradient descent: at each iteration, none of the features are never exactly zero. They are very close and one needs to threshold the feature values to actually get ones that are exactly zero. Hence the gradient is almost always well-defined.

To see that this behavior is also true when tuning the regularization parameters, we considered the lasso problem and tried tuning it. As shown below, the regularization parameter at each iteration is never exactly at the knots in the lasso path.

–Simulation goes here–