Response to Associate Editor

March 12, 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 (Zou & Hastie 2003) combines the lasso and ridge penalties; and the sparse group lasso (Simon et al. 2013) 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 (Neal 1996). From a theoretical viewpoint, multiple regularization parameters are required in certain cases to achieve oracle convergence rates. van de Geer & Muro (2014) showed that when fitting additive models with varying levels of smoothness, the penalty parameter should be smaller for more "wiggly" functions and vice versa.

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 A.4 of the Appendix.

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

   We have also included a new, very different, example illustrating the application of our ideas to matrix completion. This example moves away from the simple regression framework and considers matrix-valued data with partially observed entries (and an assumed low-rank structure). The problem now involves minimizing a penalized loss function 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) show 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 A.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 conditions specified in Theorem 1.

   The most restrictive condition is probably Condition 1, which requires that the local optimality space is also a differentiable space. In our paper, we showed that this condition is likely to hold for the lasso, group lasso, and nuclear norm penalty. We hypothesize that the condition is likely to hold more generally for non-smooth convex penalties. If we think of the penalized problem in its constrained form, the solution is the minimizer of unpenalized loss function that lies in the constraint region. For non-smooth penalty functions, the constraint region has many edges and corners; thus the constrained minimizer is likely to occur at a these non-smooth regions. Since the constrained minimizer perturbs continuously with respect to the penalty parameters, it will likely remain on the same edge/corner. As long as we remain on the same edge/corner, the optimality space and the differentiable space both stay the same and they are likely equal.

5. Make sure to provide all code for all experiments

   We have included all the code for our experiments. In addition, our code is fully available on Github at https://github.com/jjfeng/nonsmooth-joint-opt.

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?

   This is a very good question. We have added Section A.7 in the Appendix to discuss this issue in depth. We reproduce the new section below:

   > Since our algorithm depends on the validation loss being smooth almost everywhere, a potential concern is that the validation loss may not be differentiable at the solution of the joint optimization problem. We address this concern empirically. Based on the simulation study below, we suspect that the minimizer falls exactly at a knot (where our validation loss is not differentiable with respect to $\boldsymbol{\lambda}$) with measure zero.
   >
   > In this simulation we solved a penalized least squares problem with a lasso penalty and tuned the penalty parameter to minimize the loss on a separate validation set. We considered a linear model with 100 covariates. The training and validation sets included 40 and 30 observations, respectively. The response was generated data from the model
   >
   > $$y = \boldsymbol{X}\boldsymbol{\beta} + \sigma\epsilon$$
   >
   > where $\boldsymbol{\beta} = (1, 1, 1, 0, ..., 0)$. ϵ and $\boldsymbol{X}$ were drawn independently from a standard Gaussian distribution. σ was chosen so that the signal to noise ratio was 2. For a given $\lambda > 0$ our fitted $\boldsymbol{\beta}$ minimized the penalized training criterion
   >
   > $$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_T^2 + \lambda\|\boldsymbol{\beta}\|_1$$

We then chose the λ-value for which $\hat{\boldsymbol{\beta}}(\lambda)$ minimized the validation error.

In our 500 simulation runs, the penalty parameter that minimized the validation loss was never located at a knot: Using a homotopy solver for the lasso, we were able to find the *exact* knots (λ-values where variables enter/leave the model), and these points never achieved the minimum value of the validation loss. While this is only one example, and not definitive proof, we believe it is a strong indication that it is unlikely for solutions to occur regularly at knots in penalized problems.

In addition, we believe that the behavior of our procedure is analogous to solving the Lasso via sub-gradient descent. In the Lasso setting, sub-gradient descent with a properly chosen step-size will converge to the solution. In addition, if initialized at a differentiable β-value (ie. with all non-zero entries), then the lasso objective will be differentiable at all iterates in this procedure with probability one. Admittedly, using the sub-gradient method to solve the lasso has fallen out of favor. The current gold-standard methods, such as generalized gradient descent, give sparse solutions at large enough iterates and achieve faster convergence rates.

We have also added the following paragraph to Section 5:

Since our algorithm depends on the validation loss being smooth almost everywhere, a potential concern is that the validation loss may not be differentiable at the solution of the joint optimization problem. We believe that this scenario occurs with measure zero. For a more detailed discussion, refer to the Appendix.