Response to Reviewer 1

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:

Overall comment

This paper studies regularization parameter optimization via gradient descent when the training criterion is almost everywhere smooth. The authors note that the main advantage of gradient-based approaches over gradient-free approaches is their ability to handle a much larger set of regularization parameters. The methods discussed in the paper - elastic net, additive models with smoothness and sparsity penalties, and the sparse lasso - typically involve only two regularization parameters. The authors instead propose using a larger set of regularization parameters for the latter two models and present empirical results that suggest that the addition of more regularization parameters can improve performance. Although the idea of a using a larger set of regularization parameters is interesting, the empirical results are incomplete and including additional scenarios would help strengthen the paper. The following is a list of specific suggestions and comments for the authors.

We apologize for omitting simulation details and have updated Section 3 and added Section A.4 in the Appendix accordingly (we address the specific changes in the point-by-point response below). 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. We added three new sections: Section 2.4.4 introduces low-rank matrix completion and illustrates how to transform the joint optimization problem into a locally equivalent joint optimization problem; Section 3.4 provides simulation results; and 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.

Specific Suggestions/comments

1. Can the authors point to examples in the literature where a large set of regularization parameters was used?

   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. First line on p. 13, "the optimal regularization parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^\top$" should be "the optimal regularization parameters $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, ..., \lambda_M)^\top$"

   Thank you for pointing this out. We have corrected this typo in the paper.

3. The authors note that the models considered in Sections 2.4.2 and 2.4.3 are usually employed with only two regularization parameters but propose using a larger set of regularization parameters. In the corresponding simulations in Sections 3.2 and 3.3, they compare using a larger set of regularization parameters against using two regularization parameters selected using grid search. The grid spaces considered are fairly small, so its not clear if the improved performance for gradient descent is due to the additional regularization parameters or due to the dependence of performance on the grid space. To provide additional insight into whats causing the difference in performance, could the authors also present results using gradient descent, Nelder-mead, and Spearmint for the two parameter case?

   We have added the results from gradient descent, Nelder-mead, and Spearmint for the two-parameter version of the joint optimizations for all the examples. The results are all displayed in Table A.1 in the Section A.6 of the Appendix. We include the text from Section A.6 for convenience:

   > The simulation results in Section 3 show that joint optimization problems with many penalty parameters can produce better models than those with only two penalty parameters. One may wonder if this difference is due to the method used to tune the penalty parameters. Here we present results from tuning the two-penalty-parameter joint optimization problems from Sections 3.2, 3.3, and 3.4 using gradient descent, Nelder-Mead, and Spearmint. As shown in Table A.1, the performance of these methods are very similar to grid search. Regardless of the method used to tune the two-penalty parameter joint optimization, the resulting models all have higher validation and test error compared to the models from the joint optimization problem with many penalty parameters tuned by gradient descent.

4. The authors report average performance and standard errors for the simulations done in Section 3. How many simulation runs were used in each example?

   We ran each simulation example thirty times. We have added this detail to Section 3.

5. In Section 3, two different starting values were considered for Nelder-mead and gradient descent. How sensitive were the results to the choice of starting values?

   The results for Nelder-Mead and gradient descent are indeed sensitive to their starting values, but gradient descent performs better than Nelder-Mead on average. We have added Section A.5 of the Appendix to discuss the sensitivity of the two methods in more depth. We tested multiple initialization points for the two methods on a smaller version of the sparse additive model. We plot the validation error as the number of initialization points increases. The validation error of both methods plateau quickly around four initialization points. Gradient descent manages to find penalty parameters with lower validation error. Also, given a random initialization, gradient descent tends to find penalty parameters with lower validation error compared to Nelder-Mead.

6. The empirical results for gradient descent depend on α, β, and δ. The authors mention ranges considered for these parameters in Section 2.5. How were these parameters ultimately selected in the evaluations in Sections 3 and 4?

   We apologize for omitting the exact values of the gradient descent procedure. We used $\alpha = 0.001$, $\beta = 0.1$, and $\delta = 0.0005$. We have added Section A.4 in the Appendix to describe the settings used in gradient descent.

7. In Sections 3.2 and 3.3, the authors created a training, validation, and test set, but in Section 3.1 they only consider a training and validation set. Why was a test set not considered in Section 3.1?

   We apologize for the confusion. We had originally omitted the test error for Section 3.1 since the goal was to illustrate that the validation loss values were similar between gradient descent and grid search. However we recognize that this omission may be confusing to the reader. To streamline the paper, we have included a new test error column to Table 1 in Section 3.1.

8. Table 1 should note that standard errors are given in parentheses.

   We have now included this clarification to Table 1.

9. For the simulations in Section 3.3, why did the authors set n = 60 in the first case and n = 90 in the other two cases?

   The first case originally had $n = 60$ since it had fewer model parameters to estimate. In light of the reviewer's question we decided to simplify things and use $n = 90$ across all simulations in Section 3.3.

10. g is undefined in Table 4

    We have removed g from Table 4 altogether. g was meant to indicate the number of groups in the true model. However we already specify in Section 3.3 that there are three groups in the true model.

11. In Table 4, could the authors provide intuition for why there is a large difference in validation error and test error for gradient descent?

    We have added a new paragraph in Section 3 to provide some intuition for the large difference between validation and test error. We include it below:

    > In some of the examples, the models with many penalty parameters fit by gradient descent have much smaller validation errors compared to the test errors. The difference is particularly pronounced in the un-pooled sparse group lasso example in Section 3.3. There are two reasons for this behavior. First, the additional tuning parameters increase the model space and thus the degrees of freedom. Degrees of freedom relate directly to over-optimism (Tibshirani 2015). Traditionally one thinks of over-optimism as the difference between a models performance on future data and its training error. In the case of hyper-parameter tuning, performance on the validation data is the analog of the training error. The second reason is that gradient descent can effectively find a near minimizer on the validation data in contrast to Nelder-mead and Spearmint. By failing to minimize the validation data, the latter two methods is similar to ending gradient descent before it has reached convergence. This technique called early stopping is another form of regularization that can control the degree of over-optimism (Yao et al. 2007). Hence the difference between the validation and test error is greater in gradient descent compared to Nelder-mead and Spearmint.