Point-by-Point Response to Referees' Comments (2019-143)

We are grateful to the reviewers for their insightful and constructive comments. The revision has carefully addressed all the comments. The key changes are highlighted in blue in the revision. Below are our point-by-point responses to the comments.

Referee 1:

1. The reference "E. S. Kawaguchi, J. I. Shen, G. Li, and M. A. Suchard. Scalable algorithms for large competing risks data. 2019. [p]' is not clear. It is a very important reference in the paper for the algorithm details.

   Response: We thank the referee for pointing this out. The above-mentioned manuscript has been accepted pending a minor revision by the Journal of Computational and Graphical Statistics (JCGS) and we have updated the reference.

2. Fast computation of Equation 7 is the most important contribution of the paper. Most computation time is consumed on the $exp(\eta_k)$ calculation. Can the $exp(\eta_k)$ be first calculated and stored in the memory for the later computation to further improve the speed?

   Response: We note in the second paragraph on page 4 of the revision that the above mentioned strategy would work for the standard Cox's model for which only cumulative sums are involved within a risk set and the risk sets are decreasing over time. We also discuss in the third paragraph of page 4 that this strategy is, however, not sufficient and does not directly applicable to the Fine-Gray model, and that in this package, we have implemented a novel technique developed in Kawaguchi et al (2020) to reduce the computation cost from $O(n^2)$ to $O(n)$, and thus makes it scalable to large n data.

3. The figures 3 shows significant performance advantage of the fastcmprsk than other methods. However, even with very large scale data, the computation time is in the range of 10^2 seconds for the comparison methods. Can the authors discuss larger scale applications of this fast computation algorithm?

   Response: We thank the referee for their comment. We have clarified in the revision that Figure 3 is primarily used to demonstrate that " ... the computational complexity of crr increases quadratically (solid line slopes ≈ 2) while that of fastCrr is linear (dashed line slopes ≈ 1). This implies that the computational gains of fastCrr over crr are expected to grow exponentially as the sample size increases." Moreover, we have added another simulation results in Table 2 of the revision to illustrate that "...fastCrr scales well to large sample size data, whereas crr eventually grinds to a halt as n grows large. For example, for $n = 500,000$, it only takes less than 1 minute for fastCrr to finish, while crr did not finish in 3 days."

We also added in the Discussion section that " In a real-world application, Kawaguchi et al. (2019) record a drastic decrease in runtime ($\approx$ 24 hours vs. $\approx$ 30 seconds) when comparing the proposed implementation of LASSO, SCAD, and MCP to the methods available in crrp on a subset of the United States Renal Data Systems (USRDS) where $n = 125,000$."

4. In the discussion section, author mentioned that multicore is implemented for the bootstrapping calculation. Can the authors provides more details or an example for this?

   Response: Thanks for the comment! We have now included a brief explanation and example how to parallelize the variance estimation on Page 8.

   "Since standard error estimation is performed via bootstrap and resampling, it is easy to use multiple cores to speed up computation. Parallelization is seamlessly implemented using the doParallel package (Calaway et al., 2019). Enabling usage of multiple cores is done through the useMultipleCores argument within the varianceControl function. To avoid interference with other processes, we allow users to set up the cluster on their own. We provide an example below."

Referee 2:

1. The example provided in the introduction is good for demonstrating competing risk, but it is not specific for this package. A more specific example is needed to motivate the work and they need to be referred to through the manuscript.

   Response: We thank the referee for their critique. We agree that a more specific example will motivate the necessity of our package. Large-scale competing risks databases, where our methodology is expected to outperform competing methods, are typically found in electronic health record (EHR) data or cancer registry data. Our example, found in Kawaguchi et al. (2020), applies the methodology to a subset of the United States Renal Database System (USRDS) where the sample size is over $100,000$. Unfortunately, we are not authorized to make this USRDS data public and according to the **Reproducible research** section in the "Instruction for Authors" for the journal, should be not use it as an example. In any case, we have added the following in the introduction: "For example, Kawaguchi et al (2020) reported a runtime of about 24 hours to fit a LASSO regularized Fine-Gray regression on a subset of the United States Renal Data Systems (USRDS) with $n = 125,000$ subjects using an existing R package crrp". We also briefly present the conclusions of our data analysis in the Discussion section of the paper and generally expand upon the applicability of our package for large medical databases.

2. The authors make reference to an article by Kawaguchi et al. from 2019 that describes the scalable algorithms – but the reference is incomplete. Where has this been published?

   Response: The above-mentioned manuscript has been accepted pending a minor revision by JCGS and we have updated the reference.

3. Overall the preliminaries section needs to be better organized, with better transitions and more tie into the package. This section does not do a good enough job of setting up the importance for the package.

   Response: We appreciate the comment. We have renamed the previous section "Forward-backward scan for parameter estimation" to "Parameter estimation in linear time" and revised the section to facilitate a better transition to our proposed R package by demonstrating how the current approaches fail to handle large-scale competing risks data and why. Specifically, the section starts as follows on Page 3:

"Whether interest is in fitting an unpenalized model or a series of penalized models used for variable selection, one will need to minimize the negated log-pseudo (or penalized log-pseudo) likelihood. While current implementations can readily fit small to moderately-sized datasets, where the sample size can be in the hundreds to thousands, we notice that these packages grind to a halt for large-scale data such as, electronic health records (EHR) data or cancer registry data, where the number of observations easily exceed tens of thousands, as illustrated later in Section 2.5.1 (Table 2) on some simulated large competing risks data."

4. The table 1 that is presented seems incomplete – the package seems to hold more functions then what is presented (e.g., Crisk). I think that the title of the table should be adjusted to reflect what is actually being presented.

   Response: We thank the referee for pointing out the incompleteness of our table. We have revised Table 1 to include the functions that are currently available in version 1.1.0, including the S3 methods.

5. This paper is trying to demonstrate that this function can increase the speed for large data sets, but the simulations are not very large (500-4000).

   Response: Please refer to our response to comment 3 of Referee 1.

6. The code that is presented in the paper needs to be documented (use comments) and better explained in the text of the paper. For example, what do the 0, 1, and 2 event counts stand for in the output? What is presented from `round(sqrt(diag(fit3$var)),3)`?

   Response: We appreciate the comment. We apologize for the lack of explanation in our previous submission. Our revision addresses these issues, and more, by commenting the R code within the text (e.g. The highlighted R code on Pages 5-6 and 7, where we address the specific comments the referee has pointed out).

7. It would be nice if the authors could describe what the important options are for the fastCrr function and when they might be applicable.

   Response: We thank the referee for the comment. We have detailed some of the important options for the fastCrr function within the text.

   Specifically, in page 6 we added the following segment about how to modify the convergence threshold and maximum number of iterations:

“ The slight difference in numerical accuracy can be explained by the different methods of optimization and convergence thresholds used for parameter estimation. Convergence within the cyclic coordinate descent algorithm used in parameter estimation is determined by the relative change of the coefficient estimates. We allow users to modify the maximum relative change and maximum number of iterations used for optimization within the fastCrr through the eps and iter arguments, respectively. By default, we set eps = 1E-6 and iter = 1000 in both our unpenalized and penalized optimization methods. ”

We also include the following on Page 9 on what options within fastCrr would need to be specified for calculating the CIF:

“ To calculate the CIF, both the Breslow estimator of the cumulative subdistribution hazard and the (ordered) model data frame need to be returned values within the fitted object. This can be achieved by setting both the getBreslowJumps and returnDataFrame arguments within fastCrr to TRUE. ”

8. It is not clear how different the results will be from using the bootstrap method versus the original methods published by Fine and Gray.

   Response: We appreciate the response. Our previous submission highlighted the comparison between our bootstrap variance and the variance from the original crr function by observing the coverage probability of the confidence intervals produced using both variance estimates. In the revision, we have also added a simulation to compare the standard error estimates using both variance estimation methods with the empirical standard error of the coefficient estimates as an additional comparison. We include the following text on Page 12:

   “We also performed a simulation to compare the bootstrap procedure for variance estimation to the estimate of the asymptotic variance ... used in crr. First, we compare the two standard error estimates with the empirical standard error of $\hat{\boldsymbol{\beta}}_1$. For the j^{th} coefficient, the empirical standard error is calculated as the standard deviation of $\hat{\beta}_{1j}$ from the 100 Monte Carlo runs. For the standard error provided by both the bootstrap and the asymptotic variance-covariance matrix, we take the average standard error of $\hat{\beta}_{1j}$ over the 100 Monte Carlo runs. Table 3 compares the standard errors for $\hat{\beta}_{1j}$ for $j = 1, 2, 3$. When $n = 1000$, the average standard error using the bootstrap is slightly larger than the empirical standard error; whereas the standard error from the asymptotic expression is slightly smaller. These differences diminish and all three estimates are comparable when $n \geq 2000$. This provides evidence that both the bootstrap and asymptotic expression are adequate estimators of the variance-covariance expression for large datasets.

Additionally, we present in Table 4 the coverage probability (and standard errors) of the 95% confidence intervals for $\beta_{11} = 0.4$ using the bootstrap (fastCrr) and asymptotic (crr) variance estimate. The confidence intervals are wider for the bootstrap approach when compared to confidence intervals produced using the asymptotic variance estimator, especially when $n = 1000$. However, both methods are close to the nominal 95% level as n increases. We observe similar trends across the other coefficient estimates. ”

9. The varianceControl function needs to be better explained.

   Response: Thanks for the comment, we have explained the arguments that can be passed into the varianceControl function. Specifically we have added the following on Page 7:

   “ These arguments include B, the number of bootstrap samples to be used and seed, a non-negative numeric integer to set the seed for resampling. ” and have commented the necessary R code:

```
R> # Estimate variance via 100 bootstrap samples using seed 2019.
R> vc   <- varianceControl(B = 100, seed = 2019)
R> fit3 <- fastcmprsk::fastCrr(Crisk(dat$ftime, dat$fstatus) ~ Z, variance = TRUE,
+                               var.control = vc,
+                               returnDataFrame = TRUE)
# returnDataFrame = TRUE is necessary for CIF estimation (next section)

# Standard error estimates rounded to 3rd decimal place
R> round(sqrt(diag(fit3$var)), 3)

[1] 0.108 0.123 0.085 0.104 0.106 0.126 0.097 0.097 0.104 0.129
```

10. Please explain the output that is being presented from the summary calls.

    Response: We have included the following on Page 8 explaining what output is presented from the summary call:

    “ Lastly, summary will return an ANOVA table for the fitted model. The table presents the log-subdistribution hazard ratio (coef), the subdistribution hazard ratio (exp(coef)), the standard error of the log-subdistribution hazards ratio (se(coef)) if variance = TRUE in fastCrr, the corresponding z-score (z value), and two-sided p-value (Pr($|z|$)). When setting conf.int = TRUE, the summary function will also print out the 95% confidence intervals (if variance = TRUE when running fastCrr). Additionally the pseudo log-likelihood for the estimated model and the null pseudo log-likelihood (when $\hat{\boldsymbol{\beta}} = \mathbf{0}$) are also reported below the ANOVA table.

11. Several times there is mention of the run time for large sample sizes. However, relatively speaking 500-4000 is not very large, especially if one was to fit the model only once in an analysis. A stronger case needs to be made for the larger sample sizes and examples of the impact on the run time demonstrated.

    Response: Please refer to our response to Comment 3 from Referee 1.

12. The table and figure titles needs to be more descriptive. For example, figure 2 – the toy example is not describe. Tables and figure should be stand alone.

    Response: We have revised the captions for the tables and figures so that they are stand alone.

13. Please justify the use of only 100 Monte Carlo and 100 bootstrap samples.

    Response: Although we could use more than 100 Monte Carlo samples, it would be more time consuming for the simulations and unlikely to change the general conclusions. We would also like to clarify that the use of 100 bootstrap samples is only for our numerical examples and not a general recommendation. Thanks for the comment, we have added the following on page 12 (paragraph 2):

    " As shown later in the section (Tables 3 and 4), 100 bootstrap samples suffices to produce a good standard error estimate with close-to-nominal coverage for large enough sample sizes in our scenarios. In practice, we recommend users to increase the number of bootstrap samples until the variance estimate becomes stable, when computationally feasible."

    Please also see our response to comment 8.

14. Please demonstrate the statements regarding the large sample sizes and the run times. I think it is important that this be demonstrated, even if for only one data for a variety of large sample sizes from 100K to 1 million (as we might see in electronic health data).

    Response: Please see our response to Comment 3 from Referee 1.