Response Letter to the Peer-review Comments of RJournal 2021-126

Ping-Yang Chen, Hsing-Ming Chang, Yu-Ting Chen, Jung-Ying Tzeng, and Sheng-Mao Chang

We first thank the editor for handling our submission and all reviewers for their reports with very constructive and valuable comments. We have tried our best to respond to each reviewer's comment and update the manuscript and the R codes. We believe and, hopefully, you would agree that we have addressed all the main concerns, which results in an improved manuscript and a better version of the package that is more useful to the journal's audience.

1-review-2

> This article describes the R package TensorTest2D, which is used to describe the relationship between
> predictors that take the form of a matrix (i.e., a two-way array) and a univariate outcome. The package
> allows for the outcome to be continuous (normally distributed), binary, or Poisson under a GLM
> framework. The functionality of the package includes predictive modeling of the outcome and hypothesis
> testing for the coefficients. In all cases the tensor/matrix structure of the predictors is accounted for by
> a low-rank assumption, wherein the matrix of coefficients corresponding to the predictors has a specific rank.
>
> Overall, I think this package serves a useful purpose. The methods are not new – tensor regres-
> sion in a GLM / exponential family setting with low-rank constraints is a well-established approach, and
> the specific methodology for hypothesis testing of coefficients, etc. is described in a recent Bioinformatics
> article by some of the package authors (Chang, 2021). However, I am not aware of other R packages that
> can perform tensor regression for a general GLM setting, or that allow for hypothesis testing. Thus, with
> increasing relevance of tensor methods, the package fills an important gap.
>
> The manuscript is generally well-written and clearly explains the functionality of the package,
> with real-world data examples that can be reproduced. I was able to install the package from CRAN
> use it myself without issue. The documentation for the package is clear. Standard functions associated
> with predictive modeling in R – 'summary', 'plot', 'predict' – are extended to the output in an intuitive
> way. The 2D visualizations in the plot function can also be used in conjunction with other methods, as
> demonstrated by Example 2 in the manuscript.

Response: Thank you. The summary is accurate.

> Despite these strengths, I do have the following comments:
>
> 1.) Does the package need to be restricted to matrix-valued (i.e., two-way) predictors? Many
> applications involve higher-order arrays, so this is somewhat of a limitation.

Response: Thank you for pointing out this limitation. In literature, most of the Hessian matrix of arbitrary dimensional tensor regression model were derived without identifiability constraints, e.g. equation (8) of Zhou et al. (2013), and thus some manipulation is required so that the resulting Hessian matrix is invertible. On the other hand, Chang et al. (2021) derived the Hessian matrix under identifiability constraints for 2-way (matrix-valued) tensors, i.e., our package is developed in particular as a software application for omics data and 2D image data which can be conveniently represented in a matrix. This package is limited, at the moment, to matrix-valued predictors. We feel that the results in Chang et al. (2021) can likely be extended and it is worth much effort devoted in a separate discussion on compelling applications in dealing with tensor regression models with higher-order arrays as the predictor.

> 2.) The test to obtain p-values for coefficients is referred to as "Ward's test" in the manuscript (both pg 3
> and pg 4). Should this instead be "Ward's test"? If not, please provide a reference for Ward's test.

Response: Thank you for noting this. We acknowledge the typo of 'Ward's' test when it should have been the Wald test as in *Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations is Large* by Wald (1943). A correct reference for the Wald test will be included in the manuscript.

> 3.) For the coefficient summaries in the package give the 't-value' and 'Pr(>|t|)'. However, my understanding
> is that the t-distribution is not used to compute the p-value. If instead a standard normal is used, 'z-value'
> and 'Pr(>|z|)' would make more sense.

Response: Thank you very much for pointing out this problem. For a continuous response, assuming Gaussian error, the t-tests are used to test the significances of the regression coefficients. Under the generalized linear model framework, the response variable is no longer following Gaussian distribution, and the test statistics of the regression coefficients are asymptotically normal according to the theory of the Wald test. We have revised the output summary table of the `summary.tsglm` function. Given the option `family='gaussian'`, the column name of the test statistic will be "`t value`" and of the *p*-value will be "`P(>|t|)`" Otherwise, the column names will be "`z value`" and "`P(>|z|)`" respectively.

> 4.) The package includes a reference to Wu, Huang, and Ma (2017), but this article does not have enough
> information to understand the details of the methods. Please also add Chang et al., 2021 as a reference.

Response: Thank you very much for pointing out this problem. We added Chang et al. (2021) to the reference list in the package documentation.

1-review-4

> The authors here present an attractive software package that implements tensor regression using the
> PARAFAC decomposition first published by Zhou, Li, and Zhu (2013), with expanded constraints from
> Chang et al. (2021). I think this package is useful to the R community and allows users to utilize modern
> statistical methods in a user-friendly functional interface. In addition to implementing the methods outlined
> in Chang et al. (2021), the authors extend the model to binomial logistic regression and poisson log-link
> regression. They also include some extra functions to display results from the model fitting and make
> figures that are informative for judging model fit. All of the code provided in the paper ran on my computer
> without incident, and I was even able to run a model with simulated data. I have a few thoughts and
> suggestions for the paper, which I will outline below:

Response: Thank you. The summary is accurate.

> **Software Comments**
>
> 1. There are several functions created within the TRtest function itself, such as Check_tidy_input,
> VAR_ALS, %b%, %i%, etc. These functions should be written and documented outside the TRtest
> function. This also happens in the summary.tsglm function, and may happen elsewhere. Separating and
> documenting functions, even if they are not exported to the NAMESPACE, will help continue package
> maintenance and make it easier for other researchers to understand what is happening within your code.

Response: We agree to your suggestions. In the original version of our package, this issue exists in the functions `TRtest`, `summary.tsglm` and `predict.tsglm`. We have taken these sub-functions outside the main `tensorReg2D` function and written separate documents for them. Codes for the new version of our package can be found on the Github repository: https://github.com/yuting1214/TensorTest2D.

> 2. There's an inconsistent behavior in the default method of plot.tsglm. If I run the model for the omics
> data and then run plot(omicsMdl), I get a plot without a ⊠ marking significant coefficients. This does not
> match what is shown in your example. However, when I run the MNIST model and then run plot(lgMdl),
> I do see these ⊠ markings denoting significance. I think it would be best to either have the significance
> markings in the default or exclude them unless a value is defined for alpha in plot.tsglm.

Response: Thank you very much for pointing this issue out. After some thorough inspection, we have modified the default options of the `plot.tsglm` function as described in the following:

- The `method` option specifies the p-value correction method, and now its default value is `none`, which does not apply a p-value correction method.

- The `alpha` option specifies the value of significance level. Its default value is `NULL`, and hence the image plot does not show a ⊠ mark. When `alpha`$\in (0, 1]$ is specified, the image plot marks the pixels where their corresponding p-values are smaller than `alpha`.

We have updated the new version of the `plot.tsglm` function on R CRAN (please see `?plot.tsglm`) and our Github repository (https://github.com/yuting1214/TensorTest2D).

> 3. Is there a GitHub repository or other development and bug-tracking site so users can track the
> maintenance of the package and/or make suggestions? That would help the package's longevity. In
> addition, there are no unit tests for the package. While they aren't required, they do help catch bugs if the
> code is updated.

Response: Yes, our codes are available on the Github repository at https://github.com/yuting1214/TensorTest2D. For issue/bug reports, the users can post their questions on https://github.com/yuting1214/TensorTest2D/issues. We have now included these URLs in the DESCRIPTION file of our package on CRAN. Codes for unit tests based on the `testthat` package are available on our Github repository.

> **Content Comments**
>
> 1. The name of the package is somewhat confusing, and made me think that it was a test version of the package, rather than one that is fully-functioning and submitted to CRAN. I'm not sure how easy it is to change the name once it has been submitted, but taking the word "Test" out of the package title might make it more clear. This is also true for the TRtest function.

Response: We agree to your suggestion totally! We chose the word "Test' ' to highlight that our package includes the feature of statistical hypothesis testing for the coefficients of the tensor regression model. To avoid possible misunderstanding on the naming, we first changed the name of our main function `TRtest()` to be `tensorReg2D()`. As for the package name, we sent the name-changing request to the CRAN administration. As much as we would like to, but the name-changing request was unsuccessful, and we quote the original response from CRAN:

"In general, renaming of R packages is not permitted unless there are legal reasons why the old name is not feasible."

To sum up, we did try our best by coordinating with the CRAN team to change the package name at the reviewer's suggestion, we are, however, stuck with the original package name **TensorTest2D** due to CRAN's policy. In the new version of our package, the main function has a new name `tensorReg2D()`. The new version of the package is currently available on CRAN.

> **Minor comments**
>
> 1. I think it would be helpful to explain briefly why there are only three possible tensor models in the first data analysis example. I know there is an informative error message in the package itself if one tries to overspecify the rank, but a note within the paper itself would make this clearer.

Response: Thank you very much for the reminder. For a predictor matrix $\mathbf{X}$ of size $P \times G$, its maximum rank is $R \leq \min\{P, G\}$ (Chang et al., 2021). In the first data analysis example, the omics predictor is a 3×10 matrix, and therefore, the maximum rank fo the predictor matrix is 3. That is, we have three possible tensor models, namely, the rank-1, rank-2 and the rank-3 model, respectively. We have explained the reason to consider three tensor models in Example 1.

> 2. It would be nice to see the constraints imposed by Chang et al. (2021) in this paper to help understand the number of unconstrained parameters in the model as shown in this paper.

Response: Thank you for pointing this out. For clarification on the total number of unconstrained parameters, we inserted the following paragraph at the end of the Generalized Tensor Regression Model section:

Before giving a brief description of how Chang et al. (2021) place constraints on tensor regression parameterization, we wish to emphasize that the process of estimating for $\partial\mu_i/\partial\boldsymbol{\eta}$ is in typical not exactly simple. It is known that the matrix factorization (decomposition) $\mathbf{B} = \mathbf{B}_1\mathbf{B}_2^\top$ is not unique because, for every invertible matrix $\mathbf{O} \in \mathbb{R}^{R\times R}$, $\mathbf{B} = \left(\mathbf{B}_1\mathbf{O}^{-1}\right)\left(\mathbf{O}\mathbf{B_2}^\top\right)$. Write

$$\mathbf{B}_1 = \left[\begin{array}{c} \mathbf{B}_{11} \\ \mathbf{B}_{21} \end{array}\right],$$

where $\mathbf{B}_{11} \in \mathbb{R}^{R\times R}$ and $\mathbf{B}_{21} \in \mathbb{R}^{(P-R)\times R}$, and assume $\mathbf{B}_{11}$ is invertible. One way to ensure the uniqueness of the matrix factorization is to force $\mathbf{O} = \mathbf{B}_{11}$, and thus

$$\mathbf{B}_1\mathbf{B}_2^\top = \begin{bmatrix}\mathbf{B}_{11}\\ \mathbf{B}_{21}\end{bmatrix}\mathbf{O}^{-1}\mathbf{O}\mathbf{B}_2^\top = \begin{bmatrix}\mathbf{B}_{11}\mathbf{O}^{-1}\\ \mathbf{B}_{21}\mathbf{O}^{-1}\end{bmatrix}\tilde{\mathbf{B}}_2^\top = \begin{bmatrix}\mathbf{I}_R\\ \tilde{\mathbf{B}}_{21}\end{bmatrix}\tilde{\mathbf{B}}_2^\top,$$

where $\tilde{\mathbf{B}}_{21} = \tilde{\mathbf{B}}_{21}\mathbf{O}^{-1}$ and $\tilde{\mathbf{B}}_2 = \tilde{\mathbf{B}}_2\mathbf{O}^\top$. Consequently, the unknown parameter matrices are $\tilde{\mathbf{B}}_{21} \in \mathbb{R}^{(P-R)\times R}$ and $\tilde{\mathbf{B}}_2 \in \mathbb{R}^{Q\times R}$ with a total of $(P-R)\times R + Q\times R$ unknown parameters. We believe that an exact formula for $\partial\mu_i/\partial\tilde{\boldsymbol{\eta}}$ can not be found prior to the work by Chang et al. (2021) in the case when $\tilde{\boldsymbol{\eta}} = (vec(\tilde{\mathbf{B}}_{12})^\top, vec(\tilde{\mathbf{B}}_1)^\top)^\top$, and the same formula is used in **TensorTest2D**.

> 3. In the example code, the authors load the dplyr package. I think this is solely for the use of the pipe
> operator %>%, which isn’t really necessary here. If you need the pipe, you could update your depends to
> R (>= 4.1.0) and use the new native pipe |>, but I think it would probably be better to use a nested
> spacing (see below) or just define an object for the predictions to eliminate the need for a pipe.

```
> head(
>   round(
>     predict(lgMdl, test_X, type = "link"),
>     digits = 2
>   )
> )
```

Response: Thank you very much for the suggestion. We have eliminated the pipe operator and defined an object for the predictions in the example code. Please refer to Page 9.

> **Typos**
>
> 1. In the first paragraph of the introduction:
> “Compared to the existing R packages, the proposed package TensorTest2D (Chen et al., 2021) considers a
> generalized linear model (GLM) with matrix structured predictors and a scalar outcome, and it can be
> used for outcome prediction or testing.”

Response: Thank you so much for the correction. We have rephrased this sentence according to your suggestion.

> 2. A few times: Make sure you spell the name of your package correctly! I saw TesnsorTest2D instead of
> TensorTest2D more than once.

Response: Thank you for pointing out the misspelling. We have corrected all the typos.

> 3. On page 2: “The output object of the function TRtest() includes the follows”

Response: Thank you for pointing this out. We rephrase it as “The function `tensorReg2D()` returns a list object which includes the following variables’ ’. Please refer to Page 3.

> **References**
>
> Chang, Sheng-Mao, Meng Yang, Wenbin Lu, Yu-Jyun Huang, Yueyang Huang, Hung Hung, Jef-
> frey C Miecznikowski, Tzu-Pin Lu, and Jung-Ying Tzeng. 2021. “Gene-Set Integrative Analysis of
> Multi-Omics Data Using Tensor-Based Association Test.” Bioinformatics.
>
> Zhou, Hua, Lexin Li, and Hongtu Zhu. 2013. “Tensor Regression with Applications in Neuroimaging Data
> Analysis.” Journal of the American Statistical Association 108 (502): 540–52.