1 Report ASP20 Boost

First Report

Johannes Strauß Levin Wiebelt Sebastian Aristizabal

05 Juni 2020

Abstract

This is the abstract.

It consists of two paragraphs.

Contents

How to cite and configure the document's appearance

If you don't have Latex download latex for R with `tinytex::install_tinytex()`. To install pandoc on Mac, install it via Terminal with `brew install pandoc-citeproc`. ## Write

To cite:

- Parentheses: Implementation builds upon the asp20 model (Riebl 2020).
- Inline: It is implemented as R6 class Chang (2019).
- Including pages: For the theoretical foundations of our implementation, we rely mostly on (Fahrmeir 2013, 219) and (Hastie, Tibshirani, and Friedman 2009, 358).

In the `report` folder there's two `.bib` files. These contain all reference we should need. To update the `package.bib` write the name of the package in the following chunk and *actively* run it (It should be installed in your terminal). The`citavi.bib` file needs to be exported from `citavi`.

```
# automatically create a bib database for R packages
knitr::write_bib(c(
  .packages(), 'R6', 'knitr', 'rmarkdown', 'asp20model', 'gamboostLSS','mboost', 'tidyverse'
), 'packages.bib')
```

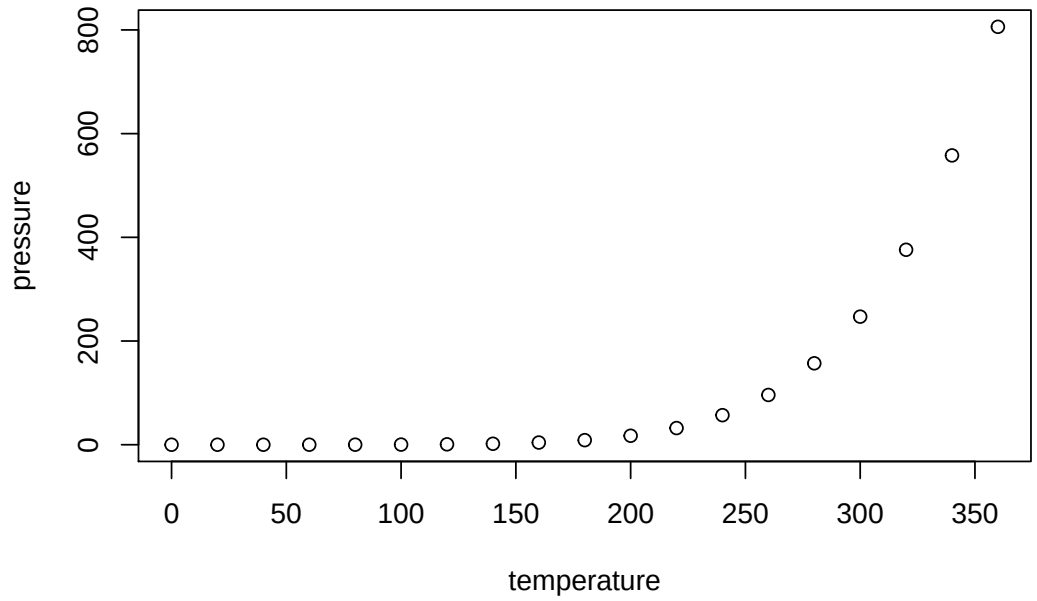
To cite write a '`@` followed by `firstname.year` as shown in the `.bib` files for reference. See further cite

As a tip hit always enter after each period for better readability in R Studio. Like this.

Two enters give me a new paragraph.

Including Plots

You can also embed plots, for example:



Appearance using the `YAML` Header:

We can regulate the appearance of the document from here. There's a lot flexibility but the learning curve is somehow steep. I'll link useful resources to learn, but, in principle, the document is *ready for writing*.

Here the corresponding references:

- PDF output
- Pandoc Manuals
- TOC
- Cite with Pandoc

Here those inputs that are easy to configuration for you guys to get an idea:

- With `documentclass` has three base templates: `article`, `report` and `book`.
- `geometry` let's me change the margins.
- `lof` and `lot` set to true give me a list of figures and tables respectively.

1. Introduction:

- The developement of this package takes place in the framework of the seminar "Advanced Statisical Programming" -> SS20
- We extend the R6 class *"LocationScaleRegression"* belonging to the `asp20model` package concieved specifficaly for this seminar (Riebl 2020)
- Our aim is to develope an multi-faceted implementation of boosting for location and scale regression including component-wise boosting, use of cross-validation to determine the optimal stopping critera and useful options for visualization.

2. Description of progress

> Hannes lead question: What are the critical aspects of your implementation and your package? Stability? Performance? User-friendliness? How are you going to deal with these potential issues?

Our package rely heavily on the asp20model package, any changes, or errors in the code of the package might break the functionality. Currently there is no validation of the given input, with respect to heteroscedasticity.

The package also does not provide any auto optimization of the gradient boost step-size, a wrongly picked step-size can lead to a deficient performance, but an already implemented early stopping mechanism can prevent an overfitting. An input check and cross-validation of the step-size is on the to-do list. Also, the initial starting values for Beta and Gamma can be further improved, see (Fahrmeir 2013, 219). Further optimizations can be made to reduce the computation time, by moving the calculation of the Projection Matrix for individual covariates in the initialization section of the model. In regard to User-friendliness of the package, we will try to implement further input validation and try to write up with a comprehensive vignette containing various examples in the next weeks.

L:

- user-friendliness: allow for convenient input of model, possibly as data.frame
- stability: trying to implement automated testing (unit tests)
- performance:
  - custom step size for Beta needed, otherwise slow –> improve
  - move ProjectionMatrix Calculation to initializalisation to enhance efficiency

2. What design designs did you make with respect to your implementation? How are you going to extend the R6 class? What additional functions are you going to provide?

JS: During the learning process, we tried several methods to implement the required functionality. At first, we started with a simple function in a R-File. While looking at other online resources, we agreed on using the asp20model packages as a requirement for our package and inheriting the existing LocationScaleRegression class. Our `LocationScaleRegressionBoost` R6 class implements a boosting algorithm, two active fields within the model for calculating the best fitting covariate with respect to Gamma and Beta and has four new variables which are set at the initialization of the model to reduce computation time of the Projection Matrixes during the gradient boosting (subject to change). The package provides the gradient boost function for the `LocationScaleRegressionBoost` class, which uses the two model functions to optimize stepwise the Beta and Gamma Vector.

I propose a timeline where the implementation process is described step by step as a list e.g:

1. Primitive implementation of boosting
2. Simple boosting for location implemented
3. Adressed the estimation for the variance of gamma.
4. Johannes finished everything #lol. 27.05.

Johannes is *gzus* ^{2}.

L: implementation designs r6-Class "LocScRegBoost"

- moved calculation of central calculation matrices in boosting algorithm to initializer of R6-Class via super-command
- component-selection implemented inside R6-Class via additional active-fields
  - mechanism: calculation of n loss-function-values (n = #gamma-parameters = #beta-parameters), then compare to loss-functions of last iteration and choose highest loss-improvement
  - calculation does NOT work via derivatives as Kneib suggested, but by Hannes' deviance-residuals

3. Description of the problem

> Hannes lead question: Which open questions do you have about the statistical model and methodology?

JS : Can this model be used to mathematically evaluate outliers, for example in the Munich rent data set -> use it to identify cheap apartments

L:

- How does cross-validation work in order to optimize stopping?
  - This connects to other applications of boosters such as variable selection. Variable selection is especially viable since component-wise boosting is already implemented
- Is it possible to optimize the learning rate?

4. Extensions to be implemented.

> Hannes' Lead question: What functionality are you planning to (or did you already) implement? What did you decide to leave out?

L: Already implemented

- boosting for multidimensional (>2) beta and gamma
- component-wise boosting

L: Ideas for further functionality

- better user-friendliness - allow inputs like my_model <- gradient_boost(formula, data = mtcars)
  - allow for model creation via dataframe-input
  - this may reduce errors due to unexpected inputs
- illustrate functions of package with real dataframe (= Vignette)
- Visualization of results via connecting to plot team `asp20plot`
- functionality: predict
- Include model diagnostics

JS: check when run gradientboost, if model is LocationScaleRegressionBoost class

5. Summary?

6. References

Chang, Winston. 2019. *R6: Encapsulated Classes with Reference Semantics*. https://CRAN.R-project.org/package=R6.

Fahrmeir, Ludwig. 2013. *Regression: Models, Methods and Applications*. New York: Springer.

Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction / Trevor Hastie, Robert Tibshirani, Jerome Friedman*. 2nd ed. Springer Series in Statistics. New York: Springer.

Riebl, Hannes. 2020. *Asp20model: An R6 Class for Location-Scale Regression Models*.