Adding Custom Data Splitting Functions

Jenna Reps

2021-12-16

Contents

1 Introduction

This vignette describes how you can add your own custom function for splitting the labelled data into training data and validation data in the Observational Health Data Sciencs and Informatics (OHDSI) `PatientLevelPrediction` package. This vignette assumes you have read and are comfortable with building single patient level prediction models as described in the `BuildingPredictiveModels` vignette.

We invite you to share your new data splitting functions with the OHDSI community through our GitHub repository.

2 Data Splitting Function Code Structure

To make a custom data splitting function that can be used within PatientLevelPrediction you need to write two different functions. The 'create' function and the 'implement' function.

The 'create' function, e.g., create<DataSplittingFunction>, takes the parameters of the data splitting 'implement' function as input, checks these are valid and outputs these as a list of class 'splitSettings' with the 'fun' attribute specifying the 'implement' function to call.

The 'implement' function, e.g., implement<DataSplittingFunction>, must take as input: * population: a data frame that contain rowId (patient identifier), ageYear, gender and outcomeCount (the class labels) * splitSettings - the output of your create<DataSplittingFunction>

The 'implement' function then needs to implement code to assign each rowId in the population to a splitId (<0 means in the train data, 0 means not used and >0 means in the training data with the value defining the cross validation fold).

3 Example

Let's consider the situation where we wish to create a split where females are used to train a model but males are used to evaluate the model.

3.1 Create function

Our gender split function requires a single parameter, the number of folds used in cross validation. Therefore create a function with a single nfold input that returns a list of class 'splitSettings' with the 'fun' attribute specifying the 'implement' function we will use.

```
createGenderSplit <- function(nfold)
  {

  # create list of inputs to implement function
  splitSettings <- list(nfold = nfold)

  # specify the function that will implement the sampling
  attr(splitSettings, "fun") <- "implementGenderSplit"

  # make sure the object returned is of class "sampleSettings"
  class(splitSettings) <- "splitSettings"
  return(splitSettings)

}
```

We now need to create the 'implement' function `implementGenderSplit()`

3.2 Implement function

All 'implement' functions for data splitting must take as input the population and the splitSettings (this is the output of the 'create' function). They must return a data.frame containing columns: rowId and index.

The index is used to determine whether the patient (identifed by the rowId) is in the test set (index = -1) or train set (index > 0). In in the train set, the value corresponds to the cross validation fold. For example, if rowId 2 is assigned index 5, then it means the patient with the rowId 2 is used to train the model and is in fold 5.

```
implementGenderSplit <- function(population, splitSettings){

  # find the people who are male:
  males <- population$rowId[population$gender == 8507]
  females <- population$rowId[population$gender == 8532]

  splitIds <- data.frame(
    rowId = c(males, females),
    index = c(
      rep(-1, length(males)),
      sample(1:splitSettings$nfold, length(females), replace = T)
    )
  )

  # return the updated trainData
  return(splitIds)
}
```

4 Acknowledgments

Considerable work has been dedicated to provide the `PatientLevelPrediction` package.

```
citation("PatientLevelPrediction")
```

```
##
## To cite PatientLevelPrediction in publications use:
##
## Reps JM, Schuemie MJ, Suchard MA, Ryan PB, Rijnbeek P (2018). "Design and implementation of a
## standardized framework to generate and evaluate patient-level prediction models using
## observational healthcare data." _Journal of the American Medical Informatics Association_,
## *25*(8), 969-975. <URL: https://doi.org/10.1093/jamia/ocy032>.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {J. M. Reps and M. J. Schuemie and M. A. Suchard and P. B. Ryan and P. Rijnbeek},
##     title = {Design and implementation of a standardized framework to generate and evaluate patient-l
##     journal = {Journal of the American Medical Informatics Association},
##     volume = {25},
##     number = {8},
##     pages = {969-975},
##     year = {2018},
##     url = {https://doi.org/10.1093/jamia/ocy032},
##   }
```

Please reference this paper if you use the PLP Package in your work:

Reps JM, Schuemie MJ, Suchard MA, Ryan PB, Rijnbeek PR. Design and implementation of a standardized framework to generate and evaluate patient-level prediction models using observational healthcare data. J Am Med Inform Assoc. 2018;25(8):969-975.

This work is supported in part through the National Science Foundation grant IIS 1251151.