DeepPatientLevelPrediction Installation Guide

Egill Fridgeirsson

2022-08-17

Contents

1 Introduction

This vignette describes how you need to install the Observational Health Data Science and Informatics (OHDSI) DeepPatientLevelPrediction under Windows, Mac and Linux.

2 Software Prerequisites

2.1 Windows Users

Under Windows the OHDSI Deep Patient Level Prediction (DeepPLP) package requires installing:

- R (https://cran.cnr.berkeley.edu/) - (R >= 3.5.0, but latest is recommended)
- Rstudio (https://www.rstudio.com/)
- Java (http://www.java.com)
- RTools (https://cran.r-project.org/bin/windows/Rtools/)

2.2 Mac/Linux Users

Under Mac and Linux the OHDSI deepPLP package requires installing:

- R (https://cran.cnr.berkeley.edu/) - (R >= 3.3.0, but latest is recommended)
- Rstudio (https://www.rstudio.com/)
- Java (http://www.java.com)
- Xcode command line tools(run in terminal: xcode-select –install) [MAC USERS ONLY]

3 Installing the Package

The preferred way to install the package is by using `remotes`, which will automatically install the latest release and all the latest dependencies.

If you do not want the official release you could install the bleeding edge version of the package (latest develop branch).

Note that the latest develop branch could contain bugs, please report them to us if you experience problems.

3.1 Installing DeepPatientLevelPrediction using remotes

To install using `remotes` run:

```r
install.packages("remotes")
remotes::install_github("OHDSI/FeatureExtraction")
remotes::install_github("OHDSI/PatientLevelPrediction")
remotes::install_github("OHDSI/DeepPatientLevelPrediction")
```

DeepPLP relies on torch for R. When torch is installed for the first time the installation is not finished until you run:

```r
library(torch)
```

This will download the required libtorch and lantern binaries for your operating system and copy them to the required locations for torch to use.

If you are using DeepPLP in an offline environment the function `torch::install_torch_from_file()` can be used. This will first require to download and move the correct binaries to the offline environment. See torch installation guide for more detailed instructions.

When installing make sure to close any other Rstudio sessions that are using `DeepPatientLevelPrediction` or any dependency. Keeping Rstudio sessions open can cause locks that prevent the package installing.

4 Testing Installation

```r
library(PatientLevelPrediction)
library(DeepPatientLevelPrediction)

data(plpDataSimulationProfile)
sampleSize <- 1e4
```

```
plpData <- simulatePlpData(
  plpDataSimulationProfile,
  n = sampleSize
)

populationSettings <- PatientLevelPrediction::createStudyPopulationSettings(
                                                          requireTimeAtRisk = F,
                                                          riskWindowStart = 1,
                                                          riskWindowEnd = 365)
# a very simple resnet
modelSettings <- setResNet(numLayers = 2,
                           sizeHidden = 64,
                           hiddenFactor = 1,
                           residualDropout = 0,
                           hiddenDropout = 0.2,
                           normalization = 'BatchNorm',
                           activation = 'RelU',
                           sizeEmbedding = 64,
                           weightDecay = 1e-6,
                           learningRate = 3e-4,
                           seed = 42,
                           hyperParamSearch = 'random',
                           randomSample = 1, device = 'cpu',batchSize = 128,
                           epochs = 3)

plpResults <- PatientLevelPrediction::runPlp(plpData = plpData,
               outcomeId = 3,
               modelSettings = modelSettings,
               analysisId = 'Test',
               analysisName = 'Testing DeepPlp',
               populationSettings = populationSettings,
               splitSettings = createDefaultSplitSetting(),
               sampleSettings = createSampleSettings(),
               featureEngineeringSettings = createFeatureEngineeringSettings(),
               preprocessSettings = createPreprocessSettings(),
               logSettings = createLogSettings(),
               executeSettings = createExecuteSettings(runSplitData = T,
                                                      runSampleData = F,
                                                      runfeatureEngineering = F,
                                                      runPreprocessData = T,
                                                      runModelDevelopment = T,
                                                      runCovariateSummary = T
                                                      ))
```

5 Acknowledgments

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

```
citation("DeepPatientLevelPrediction")
```

```
##
```

```
## To cite package 'DeepPatientLevelPrediction' in publications use:
##
##   Reps J, Fridgeirsson E, Chan You S, Kim C, John H (2021). _DeepPatientLevelPrediction: Deep
##   Learning For Patient Level Prediction Using Data In The OMOP Common Data Model_.
##   https://ohdsi.github.io/PatientLevelPrediction,
##   https://github.com/OHDSI/DeepPatientLevelPrediction.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {DeepPatientLevelPrediction: Deep Learning For Patient Level Prediction Using Data In Th
##     author = {Jenna Reps and Egill Fridgeirsson and Seng {Chan You} and Chungsoo Kim and Henrik John
##     year = {2021},
##     note = {https://ohdsi.github.io/PatientLevelPrediction, https://github.com/OHDSI/DeepPatientLevel
##   }
```

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.