Building multiple patient-level predictive models

Jenna Reps, Martijn J. Schuemie, Patrick B. Ryan, Peter R. Rijnbeek

2018-08-21

Contents

1 Introduction

This vignette describes how you can use the `PatientLevelPrediction` package to build multiple patient-level predictive models. This vignette assumes you have read and are comfortable with running single patient level prediction models.

2 Study Populations

The create a study population setting use the function `createStudyPopulationSettings`. If we want to test out three study populations:

- study population 1 includes people who have the outcome but leave the database before the end of time-at-risk and only those without the outcome who are observed for the whole time-at-risk period.
- study population 2 includes people who are observed for the whole time-at-risk period.
- study population 3 includes people who are not observed for the whole time-at-risk

we can make all three populations and then combine them into a list:

```
studyPop1 <- createStudyPopulationSettings(binary = T,
                                           includeAllOutcomes = T,
                                           removeSubjectsWithPriorOutcome = TRUE,
                                           priorOutcomeLookback = 99999,
                                           requireTimeAtRisk = T,
                                           minTimeAtRisk=364,
                                           riskWindowStart = 1,
                                           riskWindowEnd = 365,
                                           verbosity = "INFO")
studyPop2 <- createStudyPopulationSettings(binary = T,
                                           includeAllOutcomes = F,
                                           removeSubjectsWithPriorOutcome = TRUE,
                                           priorOutcomeLookback = 99999,
                                           requireTimeAtRisk = T,
                                           minTimeAtRisk=364,
                                           riskWindowStart = 1,
                                           riskWindowEnd = 365,
                                           verbosity = "INFO")
studyPop3 <- createStudyPopulationSettings(binary = T,
                                           includeAllOutcomes = F,
                                           removeSubjectsWithPriorOutcome = TRUE,
                                           priorOutcomeLookback = 99999,
                                           requireTimeAtRisk = F,
                                           minTimeAtRisk=364,
                                           riskWindowStart = 1,
                                           riskWindowEnd = 365,
                                           verbosity = "INFO")

populationSettingList <- list(studyPop1,studyPop2,studyPop3)
populationSettingNames <- c('all outcomes but only non-outcomes with complete TAR',
                            'only include people who are observed for all TAR',
                            'including people who leave during TAR')
```

3 Covariate Settings

The covariate settings are created using `createCovariateSettings`. We create seperate covariate settings and then combine them into a list:

```
covSet1 <- createCovariateSettings(useDemographicsGender = T,
                                   useDemographicsAgeGroup = T,
                                   useConditionGroupEraAnyTimePrior = T,
                                   useDrugGroupEraAnyTimePrior = T)
covSet2 <- createCovariateSettings(useDemographicsGender = T,
                                   useDemographicsAgeGroup = T,
                                   useConditionGroupEraAnyTimePrior = T,
                                   useDrugGroupEraAnyTimePrior = F)
covariateSettingList <- list(covSet1, covSet2)
covariateSettingNames < c('all demo, condition/drug groups','all demo, condition groups')
```

4 Model Settings

The model settings requires running the setModel functions for the machine learning models of interest and specifying the hyper-parameter search and then combining these into a list. For example, if we wanted to try a logistic regression, gradient boosting machine and ada boost model then:

```
gbm <- setGradientBoostingMachine()
lr <- setLassoLogisticRegression()
ada <- setAdaBoost()

modelList <- list(gbm, lr, ada)
modelSettingNames < c('default gradient boosting','default lasso lr', 'default ada boost')
```

5 Creating the model analysis list

To create the complete plp model settings use `createPlpModelSettings` to combine the population, covariate and model settings.

```
modelAnalysisList <- createPlpModelSettings(modelList = modelList,
                                   covariateSettingList = covariateSettingList,
                                   populationSettingList = populationSettingList,
                                   modelSettingNames = modelSettingNames,
                                   covariateSettingNames = covariateSettingNames,
                                   populationSettingNames = populationSettingNames)
```

6 Running the multiple prediction patient-level-prediction

As we will be downloading loads of data in the multiple plp analysis it is useful to set fftempdir to a directory with write access and plenty of space. `options(fftempdir = 'T:/fftemp')`

To run the study requires setting up a connectionDetails object

```
dbms <- "your dbms"
user <- "your username"
pw <- "your password"
```

```
server <- "your server"
port <- "your port"

connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = dbms,
                                                                 server = server,
                                                                 user = user,
                                                                 password = pw,
                                                                 port = port)
```

Next you need to specify the cdmDatabaseSchema where your cdm database is found and workDatabaseSchema where your target population and outcome cohorts are.

```
cdmDatabaseSchema <- "your cdmDatabaseSchema"
workDatabaseSchema <- "your workDatabaseSchema"
```

Now you can run the multiple patient-level prediction analysis by specifying the target cohort ids and outcome ids

```
allresults <- runPlpAnalyses(connectionDetails = connectionDetails,
                             cdmDatabaseSchema = cdmDatabaseSchema,
                             oracleTempSchema = cdmDatabaseSchema,
                             cohortDatabaseSchema = workDatabaseSchema,
                             cohortTable = "your cohort table",
                             outcomeDatabaseSchema = workDatabaseSchema,
                             outcomeTable = "your cohort table",
                             cdmVersion = 5,
                             outputFolder = "./PlpMultiOutput",
                             modelAnalysisList = modelAnalysisList,
                             cohortIds = c(2484,6970),
                             cohortNames = c('visit 2010','test cohort'),
                             outcomeIds = c(7331,5287),
                             outcomeNames =  c('outcome 1','outcome 2'),
                             maxSampleSize = NULL,
                             minCovariateFraction = 0,
                             normalizeData = T,
                             testSplit = "person",
                             testFraction = 0.25,
                             splitSeed = NULL,
                             nfold = 3,
                             verbosity = "INFO")
```

This will then save all the plpData objects from the study into "./PlpMultiOutput/plpData", the populations for the analysis into "./PlpMultiOutput/population" and the results into "./PlpMultiOutput/Result". The csv named settings.csv found in "./PlpMultiOutput" has a row for each prediction model developed and points to the plpData and popualtion used for the model development, it also has descriptions of the cohorts and settings if these are input by the user.

7 Loading the multiple prediction patient-level-prediction results

ToDo

8 Viewing the multiple prediction patient-level-prediction results

To view the results for the multiple prediction analysis, load the results and then run...