EvaluatingPhenotypeAlgorithms

Joel N. Swerdel

2020-07-25

Contents

1 Introduction

The `Phevaluator` package enables evaluating the performance characteristics of phenotype algorithms (PAs) using data from databases that are translated into the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM).

This vignette describes how to run the PheValuator process from start to end in the `Phevaluator` package.

2 Overview of Process

There are several steps in performing a PA evaluation: 1. Creating the extremely specific (xSpec), extremely sensitive (xSens), and prevalence cohorts 2. Creating the Diagnostic Predictive Model and the Evaluation Cohort using the PatientLevelPrediction (PLP) package 5. Evaluating the PAs 6. Examining the results of the evaluation

Each of these steps is described in detail below. For this vignette we will describe the evaluation of PAs for diabetes mellitus (DM).

2.1 Creating the Extremely Specific (xSpec), Extremely Sensitive (xSens), and Prevalence Cohorts

The extremely specific (xSpec), extremely sensitive (xSens), and prevalence cohorts are developed using the ATLAS tool. The xSpec is a cohort where the subjects in the cohort are likely to be positive for the health outcome of interest (HOI) with a very high probability. This may be achieved by requiring that subjects have multiple condition codes for the HOI in their patient record. An example of this for DM is included in the OHDSI ATLAS repository. In this example each subject has an initial condition code for DM. The cohort definition further specifies that each subject also has a second code for DM between 1 and 30 days after the initial DM code and 10 additional DM codes in the rest of the patient record. This very specific algorithm for DM ensures that the subjects in this cohort have a very high probability for having the condition of DM. This PA also specifies that subjects are required to have at least 365 days of observation in their patient record.

An example of an xSens cohort is created by developing a PA that is very sensitive for the HOI. The system uses the xSens cohort to create a set of "noisy" negative subjects, i.e., subjects with a high likelihood of not having the HOI. This group of subjects will be used in the model building process and is described in detail below. An example of an xSens cohort for DM is also in the OHDSI ATLAS repository.

The system uses the prevalence cohort to provide a reasonable approximation of the prevalence of the HOI in the population. This improves the calibration of the predictive model. The system will use the xSens cohort as the default if a prevalence cohort is not specified. This group of subjects will be used in the model building process and is described in detail below. An example of an prevalence cohort for DM is also in the OHDSI ATLAS repository.

2.2 Evaluating phenotype algorithms for health conditions

The function createEvaluationCohort creates a diagnostic predictive model and an evaluation cohort that will allow the user to perform an analysis for determining the performance characteristics for one or more phenotype algorithms (cohort definitions) for health conditions. This function initiates the process for the first two steps in PheValuator, namely:
1) Develop a diagnostic predictive model for the health condition.
2) Select a large, random set of subjects from the dataset and use the model to determine the probability of each of the subjects having the health condition.

createEvaluationCohort should have as inputs:

- connectionDetails - connectionDetails created using the function createConnectionDetails in the DatabaseConnector package
- oracleTempSchema - A schema where temp tables can be created in Oracle (default==NULL)

- xSpecCohortId - The number of the "extremely specific (xSpec)" cohort definition id in the cohort table (for noisy positives)
- xSensCohortId - The number of the "extremely sensitive (xSens)" cohort definition id in the cohort table (used to estimate population prevalence and to exclude subjects from the noisy positives)
- prevalenceCohortId - The number of the cohort definition id to determine the disease prevalence, usually a super-set of the exclCohort
- cdmDatabaseSchema - The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
- cohortDatabaseSchema - The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. If cohortTable = DRUG_ERA, cohortDatabaseSchema is not used by assumed to be cdmSchema. Requires read permissions to this database.
- cohortTable - The tablename that contains the at risk cohort. If cohortTable <> DRUG_ERA, then expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
- workDatabaseSchema - The name of a database schema where the user has write capability. A temporary cohort table will be created here.
- covariateSettings - There are two choices for this setting depending on the type of halth outcome to be analyzed:
    1) For chronic health outcomes, supply the function **createDefaultChronicCovariateSettings()** with the parameters for this function being:
    a) excludedCovariateConceptIds - A list of conceptIds to exclude from featureExtraction which should include all concept_ids used to create the xSpec and xSens cohorts
    b) addDescendantsToExclude - Should descendants of excluded concepts also be excluded? (default=FALSE)

    2) For acute health outcomes, supply the function **createDefaultAcuteCovariateSettings()** with the parameters for this function the same as in the function for chronic health conditions, namely:
    a) excludedCovariateConceptIds - A list of conceptIds to exclude from featureExtraction which should include all concept_ids used to create the xSpec and xSens cohorts
    b) addDescendantsToExclude - Should descendants of excluded concepts also be excluded? (default=FALSE)

- mainPopulationCohortId - The number of the cohort to be used as a base population for the model (default=0)
- mainPopulationCohortIdStartDay - When specifying a mainPopulationCohortId, the number of days relative to the mainPopulationCohortId cohort start date to begin including visits. (default=0, i.e., 0 days before the start of the mainPopulationCohortId cohort start date)
- mainPopulationCohortIdEndtDay - When specifying a mainPopulationCohortId, the number of days relative to the mainPopulationCohortId cohort start date to end including visits. (default=0, i.e., 0 days after the start of the mainPopulationCohortId cohort start date)
- baseSampleSize - The maximum number of subjects in the evaluation cohort (default=2M)
- lowerAgeLimit - The lower age for subjects in the model (default=NULL)
- upperAgeLimit - The upper age for subjects in the model (default=NULL)
- visitLength - The minimum length of index visit for noisy negative comparison for acute health conditions (default=3 days). As a guideline, use the average length of a hospital visit for those with the health condition.
- gender - The gender(s) to be included (default c(8507, 8532))
- startDate - The starting date for including subjects in the model (default=NULL)
- endDate - The ending date for including subjects in the model (default=NULL)
- cdmVersion - The CDM version of the database (default=5)
- outFolder - The folder where the output files will be written (default=working directory)
- evaluationCohortId - A string designation for the evaluation cohort file (default = "main")

- removeSubjectsWithFutureDates - Should observation end dates be checked to remove future dates (default=TRUE)
- savePlpData - Determines whether the large PLP data file is saved. Setting this to FALSE will reduce the use of disk space (default=FALSE)
- modelType - The type of health condition to be analyzed either "chronic" (default) or "acute"

The createEvaluationCohort function will produce the following artifacts:
1) A Patient Level Prediction file (in .rds format) containg the information from the model building process
2) A Patient Level Prediction file (in .rds format) containg the information from applying the model to the evaluation cohort

For example:

```
options(fftempdir = "c:/temp/ff") #place to store large temporary files

connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/ohdsi",
                                             user = "joe",
                                             password = "supersecret")

phenoTest <- createEvaluationCohort(connectionDetails = connectionDetails,
                                    xSpecCohortId = 1769699,
                                    xSensCohortId = 1770120,
                                    prevalenceCohortId = 1770119,
                                    cdmDatabaseSchema = "my_cdm_data",
                                    cohortDatabaseSchema = "my_results",
                                    cohortTable  = "cohort",
                                    workDatabaseSchema = "scratch.dbo",
                                    covariateSettings =
                                     createDefaultChronicCovariateSettings(
                                      excludedCovariateConceptIds = c(201826),
                                      addDescendantsToExclude = TRUE),
                                    baseSampleSize = 2000000,
                                    lowerAgeLimit = 18,
                                    upperAgeLimit = 90,
                                    gender = c(8507, 8532),
                                    startDate = "20101010",
                                    endDate = "21000101",
                                    cdmVersion = "5",
                                    outFolder = "c:/phenotyping",
                                    evaluationCohortId = "diabetes",
                                    removeSubjectsWithFutureDates = TRUE,
                                    saveEvaluationCohortPlpData = FALSE,
                                    modelType = "chronic")
```

In this example, we used the cohorts developed in the "my_results" cdm, specifying the location of the cohort table (cohortDatabaseSchema, cohortTable - "my_results.cohort") and where the model will find the conditions, drug exposures, etc. to inform the model (cdmDatabaseSchema - "my_cdm_data"). The subjects included in the model will be those whose first visit in the CDM is between January 1, 2010 and December 31, 2017. We are also specifically excluding the concept ID 201826, "Type 2 diabetes mellitus", which was used to create the xSpec cohort as well as all of the descendants of that concept ID. Their ages at the time of first visit will be between 18 and 90.

In this example, the parameters specify that the function will create the model file:
"c:/phenotyping/model_diabetes.rds",

produce the evaluation cohort file:

"c:/phenotyping/evaluationCohort_diabetes.rds"

2.3 Evaluating the phenotype algorithms to be used in studies

The function testPhenotypeAlgorithm allows the user to determine the performance characteristics of phenotype algorithms (cohort defintions) to be used in studies. It uses the evaluation cohort developed in the previous step. The same evaluation cohort may be used to test as many different phenotype algorithms as you wish that pertain to the same health condition.

testPhenotypeAlgorithm should the following parameters:

- connectionDetails - connectionDetails created using the function createConnectionDetails in the DatabaseConnector package
- cutPoints - A list of threshold predictions for the evaluations. Include "EV" for the expected value
- outFolder - The folder where the cohort evaluation output files are written
- evaluationCohortId - A string used to generate the file names for the evaluation cohort. This will be the same name used in createEvaluationCohort described above.
- cdmDatabaseSchema - The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specifiy both the database and the schema, so for example 'cdm_instance.dbo'.
- cohortDatabaseSchema - The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. Requires read permissions to this database.
- cohortTable - The tablename that contains the at risk cohort. The expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
- phenotypeCohortId - The ID of the cohort to evaluate in the specified cohort table.
- washoutPeriod - The mininum required continuous observation time prior to index date for subjects within the cohort to test (Default = 0). For example:

```
options(fftempdir = "c:/temp/ff") #place to store large temporary files

connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             server = "localhost/ohdsi",
                                             user = "joe",
                                             password = "supersecret")

phenotypeResults <- testPhenotypeAlgorithm(connectionDetails,
                                    cutPoints = c("EV"),
                                    outFolder = "c:/phenotyping",
                                    evaluationCohortId = "diabetes",
                                    phenotypeCohortId = 7142,
                                    cdmDatabaseSchema = "my_cdm_data",
                                    cohortDatabaseSchema = "my_results",
                                    cohortTable  = "cohort",
                                    washoutPeriod = 365)
```

In this example, we are using only the expected value ("EV"). Given that parameter setting, the output from this step will provide performance characteristics (i.e, sensitivity, specificity, etc.) at each prediction threshold as well as those using the expected value calculations as described in the Step 2 diagram. The evaluation uses the prediction information for the evaluation cohort developed in the prior step. This function returns a dataframe with the performance characteristics of the phenotype algorithm that was tested. The user can write this dataframe to a csv file using code such as:

```
      write.csv(phenotypeResults, "c:/phenotyping/diabetes_results.csv", row.names = FALSE)
```