An example for Frequent Pattern Mining using the Eunomia package

11/12/2020

Contents

0.0.1 Connect to the database

```
### Define database parameters
cdmdatabaseschema = "main"
resultsdatabaseschema = "main"
fpm_inputFile = "fpm_testing.txt"
fpm_outputFile_SPAM = "fpm_testingResults_SPAM.txt"
fpm_outputFile_SPADE = "fpm_testingResults_SPADE.txt"
fpm_outputFile_prefixSpan = "fpm_testingResults_prefixSpan.txt"
fpm_outputFile_Clasp = "fpm_testingResults_Clasp.txt"
fpm_outputFile_CMClasp = "fpm_testingResults_CMClasp.txt"
fpm_outputFile_MaxSP = "fpm_testingResults_MaxSP.txt"
fpm_outputFile_VMSP = "fpm_testingResults_VMSP.txt"
fpm_outputFile_VGEN = "fpm_testingResults_VGEN.txt"
fpm_outputFile_RuleGrowth = "fpm_testingResults_RuleGrowth.txt"
fpm_outputFile_ERMiner =  "fpm_testingResults_ERMiner.txt"

connectionDetails <- Eunomia::getEunomiaConnectionDetails()
connection <- connect(connectionDetails)
#on.exit(DatabaseConnector::disconnect(connection)) #Close db connection on error or exit
```

0.0.2 Define cohort

```
# Define cohort
cohort <- "SELECT person_id AS subject_id,
  condition_start_date AS cohort_start_date
INTO #diagnoses
FROM @cdm.condition_occurrence
WHERE condition_concept_id IN (
    SELECT descendant_concept_id
    FROM @cdm.concept_ancestor
    WHERE ancestor_concept_id = 4329847 -- Myocardial infarction
)
  AND condition_concept_id NOT IN (
    SELECT descendant_concept_id
    FROM @cdm.concept_ancestor
    WHERE ancestor_concept_id = 314666 -- Old myocardial infarction
);
INSERT INTO @cdm.cohort (subject_id, cohort_start_date, cohort_definition_id)
SELECT subject_id,
  cohort_start_date,
  CAST (1 AS INT) AS cohort_definition_id
FROM #diagnoses
INNER JOIN @cdm.visit_occurrence
  ON subject_id = person_id
    AND cohort_start_date >= visit_start_date
    AND cohort_start_date <= visit_end_date
WHERE visit_concept_id IN (9201, 9203, 262); -- Inpatient or ER;"

renderTranslateExecuteSql(connection, cohort, cdm = cdmdatabaseschema)

sql <- "ALTER TABLE #diagnoses ADD cohort_definition_id INT NOT NULL DEFAULT(1)"

# Execute the script to receive the data
renderTranslateExecuteSql(connection, sql)

querySql(connection, "SELECT count(*) FROM diagnoses;")
```

0.0.3 Get the data and close the connection

```
# Define covariate settings
TemporalcovariateSettings_eunomia <- createTemporalCovariateSettings(useConditionOccurrence = TRUE,
                                                                      temporalStartDays = seq(-(60*365), -1, by = 1) ,
                                                                      temporalEndDays = seq(-(60*365)+1, 0, by = 1))

# Extract covariates
TemporalcovariateData_eunomia <- getDbCovariateData(connection = connection,
                          cdmDatabaseSchema = cdmdatabaseschema,
                          cohortDatabaseSchema = resultsdatabaseschema,
                          cohortTable = "diagnoses",
                          rowIdField = "subject_id",
                          covariateSettings = TemporalcovariateSettings_eunomia,
```

```
                          cohortTableIsTemp = TRUE)


disconnect(connection)
```

0.0.3.1 Frequent pattern mining

0.1 Prepare the data

```
testData <- getInputFileForFrequentPatterns(covariateDataObject = TemporalcovariateData_eunomia, fileTo
```

```
## Extracting temporal data...

## Extracting covariate names...

## Generating input file for frequent pattern mining...

## Input data has been created succesfully and saved in fpm_testing.txt
```

0.2 Run SPAM

```
spam_frequentPatterns <- runFrequentPatterns(algorithm = "SPAM",
                                             inputFile = fpm_inputFile,
                                             outputFile = fpm_outputFile_SPAM,
                                             minsup = 0.5,
                                             showID = TRUE)
head(spam_frequentPatterns)
```

0.3 Run SPADE

```
spade_frequentPatterns <- runFrequentPatterns(algorithm = "SPADE",
                                             inputFile = fpm_inputFile,
                                             outputFile = fpm_outputFile_SPADE,
                                             minsup = 0.5,
                                             showID = TRUE)

head(spade_frequentPatterns)
```

0.4 Run prefixSpan

```
pS_frequentPatterns <- runFrequentPatterns(algorithm = "prefixSpan",
                                           inputFile = fpm_inputFile,
                                           outputFile = fpm_outputFile_prefixSpan,
                                           minsup = 0.5,
                                           showID = TRUE)

head(pS_frequentPatterns)
```

0.5 Run Clasp

```
clasp_frequentPatterns <- runFrequentPatterns(algorithm = "Clasp",
                                              inputFile = fpm_inputFile,
                                              outputFile = fpm_outputFile_Clasp,
                                              minsup = 0.50,
                                              showID = TRUE )

head(clasp_frequentPatterns)
```

0.6 Run CM-Clasp

```
cmclasp_frequentPatterns <- runFrequentPatterns(algorithm = "CM-Clasp",
                                              inputFile = fpm_inputFile,
                                              outputFile = fpm_outputFile_CMClasp,
                                              minsup = 0.50,
                                              showID = TRUE )

head(cmclasp_frequentPatterns)
```

0.7 Run VMSP

```
vmsp_frequentPatterns <- runFrequentPatterns(algorithm = "VMSP",
                                              inputFile = fpm_inputFile,
                                              outputFile = fpm_outputFile_VMSP,
                                              minsup = 0.50,
                                              showID = TRUE )

head(vmsp_frequentPatterns)
```

0.8 Run VGEN

```
vgen_frequentPatterns <- runFrequentPatterns(algorithm = "VGEN",
                                              inputFile = fpm_inputFile,
                                              outputFile = fpm_outputFile_VGEN,
                                              minsup = 0.50,
```

```
                                            showID = TRUE )

head(vgen_frequentPatterns)
```

0.9 Run RuleGrowth

```
ruleGrowth_frequentPatterns <- runFrequentPatterns(algorithm = "RuleGrowth",
                                            inputFile = fpm_inputFile,
                                            outputFile = fpm_outputFile_RuleGrowth,
                                            minsup = 0.50,
                                            minconf = 0.50,
                                            showID = FALSE #Does not retrieve IDs
                                            )

head(ruleGrowth_frequentPatterns)
```

0.10 Run RuleGrowth

```
erminer_frequentPatterns <- runFrequentPatterns(algorithm = "ERMiner",
                                            inputFile = fpm_inputFile,
                                            outputFile = fpm_outputFile_ERMiner,
                                            minsup = 0.50,
                                            minconf = 0.5,
                                            showID = TRUE #Does not retrieve IDs
                                            )

head(erminer_frequentPatterns)
```