Overview of the CohortMethod package

Martijn Schuemie
## CohortMethod is part of the OHDSI Methods Library

<table>
<thead>
<tr>
<th>Method Characterization</th>
<th>Estimation Methods</th>
<th>Prediction Methods</th>
<th>Supporting Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cohort Method</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New-user cohort studies using large-scale regression for propensity and outcome models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Self-Controlled Case Series</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-Controlled Case Series analysis using few or many predictors, includes splines for age and seasonality.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Self-Controlled Cohort</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A self-controlled cohort design, where time preceding exposure is used as control.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IC Temporal Pattern Disc.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A self-controlled design, but using temporal patterns around other exposures and outcomes to correct for time-varying confounding.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Case-control</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case-control studies, matching controls on age, gender, provider, and visit date. Allows nesting of the study in another cohort.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Patient Level Prediction</strong></th>
<th><strong>Feature Extraction</strong></th>
<th><strong>Empirical Calibration</strong></th>
<th><strong>Method Evaluation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Build and evaluate predictive models for user-specified outcomes, using a wide array of machine learning algorithms.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically extract large sets of features for user-specified cohorts using data in the CDM.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use negative control exposure-outcome pairs to profile and calibrate a particular analysis design.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use real data and established reference sets as well as simulations injected in real data to evaluate the performance of methods.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Database Connector</strong></th>
<th><strong>Sql Render</strong></th>
<th><strong>Cyclops</strong></th>
<th><strong>Ohdsi R Tools</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect directly to a wide range of database platforms, including SQL Server, Oracle, and PostgreSQL.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate SQL on the fly for the various SQL dialects.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highly efficient implementation of regularized logistic, Poisson and Cox regression.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support tools that didn’t fit other categories, including tools for maintaining R libraries.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Technologies

*CohortMethod* uses

- *DatabaseConnector* and *SqlRender* to interact with the CDM data
  - SQL Server
  - Oracle
  - PostgreSQL
  - Amazon RedShift
  - Microsoft APS
- *ff* to work with large data objects
- *Cyclops* for large scale regularized regression
Graham study steps

1. Getting the necessary data from the database
2. Defining the study population
3. Creating a propensity model
4. Matching
5. Fitting the outcome model

+ generating various diagnostics
Generic study steps

1. Getting the necessary data from the database
2. Defining the study population
3. [Creating a propensity model]
4. [Trimming / Matching / Stratification]
5. Fitting the outcome model

+ generating various diagnostics
Replication of Garbe et al. using the OHDSI framework

DOI 10.1007/s00228-012-1334-2

PHARMACOEPIDEMIOLGY AND PRESCRIPTION

High-dimensional versus conventional propensity scores in a comparative effectiveness study of coxibs and reduced upper gastrointestinal complications

E. Garbe · S. Kloss · M. Suling · I. Pigeot · S. Schneeweiss

Received: 28 February 2012 / Accepted: 6 June 2012 / Published online: 5 July 2012
© Springer-Verlag 2012
What is the design used by Garbe et al?

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Design choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target cohort (T)</td>
<td>Celecoxib new users</td>
</tr>
<tr>
<td>Comparator cohort (C)</td>
<td>Traditional non-steroid antiflammatory drugs (NSAID) new users</td>
</tr>
<tr>
<td>Outcome cohort (O)</td>
<td>Upper gastrointestinal complications (UGIC)</td>
</tr>
<tr>
<td>Time-at-risk</td>
<td>cohort start → cohort end</td>
</tr>
<tr>
<td>Model specification</td>
<td>1:1 propensity score-matched multivariable conditional Poisson regression</td>
</tr>
</tbody>
</table>
Step 1: Getting the necessary data from the database
Step 1: Getting the necessary data from the database

• Target, comparator, and outcome cohorts
  – From the $cohort$ table in the CDM (ATLAS)
  – From a table with the same structure as the $cohort$ table
  – From the $drug\_era$ and/or $condition\_era$ tables
  – CohortMethod can
    • limit to first exposure
    • remove subjects in both cohorts
    • enforce washout period

• Covariates
  – Automatically constructed default set
  – Custom defined covariates (see $FeatureExtraction$ package)
  – Need to exclude drugs of interest (done automatically when using $drug\_era$)
getDbCohortMethodData

Arguments for connecting to the database:

- **connectionDetails**: How to connect to the database
- **cdmDatabaseSchema**: The database schema of the CDM
- **oracleTempSchema**: Only used on Oracle
- **cdmVersion**: currently 4 or 5 are supported
getDbCohortMethodData

Arguments for finding the exposures:

- **exposureDatabaseSchema**: Database schema of exposures
- **exposureTable**: Table of exposures
- **targetId**: Cohort definition ID or drug concept ID
- **comparatorId**: Cohort definition ID or drug concept ID
- **firstExposureOnly**: restrict to first exposure per person
- **removeDuplicateSubjects**: remove subjects in both cohorts
- **washoutPeriod**: enforce minimum amount of observation prior to index
- **studyStartDate, studyEndDate**: Also truncates follow-up time
getDbCohortMethodData

Arguments for finding the outcomes:

- **outcomesDatabaseSchema**: Database schema of outcomes
- **outcomesTable**: Table of outcomes
- **outcomeIds**: Cohort definition IDs or condition concept IDs
getDbCohortMethodData

Arguments for creating the covariates:

• **covariateSettings**: Created using the covariateSettings function

• **excludeDrugsFromCovariates**: Automatically exclude drugs of interest from the covariates (only works if targetId and comparatorId are concept IDs)
getDbCohortMethodData

Result:

An object of type cohortMethodData

Need to save and load using saveCohortMethodData and loadCohortMethodData
Diagnostics

Run `summary()` on cohortMethodData object

- Do target, comparator, and outcomes have subjects?
- Are covariates constructed?

CohortMethodData object summary

Treatment concept ID: 1
Comparator concept ID: 2
Outcome concept ID(s): 3

Treated persons: 17058
Comparator persons: 13566

Outcome counts:
  Event count Person count
  3     6535     4279

Covariates:
Number of covariates: 17
Number of non-zero covariate values: 70605
Now try it yourself!

- Go to [http://hix.jnj.com/atlas/#/estimation/5](http://hix.jnj.com/atlas/#/estimation/5)
- Click on *Export* and then *R Code*
- Run the *library* commands
- Specify the connection details...
Now try it yourself!

- Specify the connection details...
- Run commands up to and including `saveCohortMethodData` command
- Run `summary` on cohortMethodData object
- Did everything go ok?
CohortMethodData object summary

Treatment concept ID: 1
Comparator concept ID: 2
Outcome concept ID(s): 3

Treated persons: 17058
Comparator persons: 13566

Outcome counts:
  Event count Person count
  3       6535     4279

Covariates:
Number of covariates: 17
Number of non-zero covariate values: 70605
Step 2: Defining the study population
Step 2: Defining the study population

- Select one of the outcomes of interest
- Enforce additional filtering criteria
- Define risk window
createStudyPopulation

Misc arguments:

- **cohortMethodData**: As created using `getDbCohortMethodData`
- **outcomeId**: The ID of the outcome of interest
- **firstExposureOnly, removeDuplicateSubjects, washoutPeriod**: Same as in `getDbCohortMethodData`
- **removeSubjectsWithPriorOutcome**: Remove subjects who have the outcome prior to the index date?
- **priorOutcomeLookback**: How many days should we look back
createStudyPopulation

Arguments for risk window:

• **riskWindowStart**: Start day relative to index
• **addExposureDaysToStart**: Set to TRUE if riskWindowStart should be relative to exposure end date instead
• **riskWindowEnd**: End day relative to index
• **addExposureDaysToEnd**: Set to TRUE if riskWindowEnd should be relative to exposure end date instead
• **minDaysAtRisk**: Remove subjects with less than this number of days at risk

Risk window = time exposed
riskWindowStart = 0
addExposureDaysToStart = FALSE
riskWindowEnd = 0
addExposureDaysToEnd = TRUE

Risk window = intent to treat
riskWindowStart = 0
addExposureDaysToStart = FALSE
riskWindowEnd = 9999
addExposureDaysToEnd = FALSE

Risk window = time exposed + 30 days
riskWindowStart = 0
addExposureDaysToStart = FALSE
riskWindowEnd = 30
addExposureDaysToEnd = TRUE

Risk window = 30 days following index
riskWindowStart = 0
addExposureDaysToStart = FALSE
riskWindowEnd = 30
addExposureDaysToEnd = FALSE
createStudyPopulation

Result:

A data frame specifying the study population
Diagnostics

run `getAttritionTable` or `drawAttritionDiagram`

• Are the number of dropouts what you’d expect?

Original cohorts:
  Treated: n = 19046
  Comparator: n = 50918

No prior outcome
  N
  Treated: n = 1198
  Comparator: n = 4809

Have at least 1 days at risk
  N
  Treated: n = 82
  Comparator: n = 447

Study population:
  Treated: n = 17766
  Comparator: n = 45662
Now try it yourself!

- Run code in section *Defining the study population*
- Check the attrition diagram

This is what you should get:
Step 3: Creating a propensity model
Step 3: Creating a propensity model

Using regularized logistic regression

\[ P(\text{treatment} \mid X) = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots) \]

With prior for every \( \beta \):
Hyper-parameter

What is the right width of the prior distribution?

Too wide:
- convergence problems
- overfitting

Too narrow:
- ‘underfitting’: missing important predictors

Default: select hyper-parameter through 10-fold cross-validation. This aims to optimize the out-of-sample likelihood
createPs

Important arguments:

- **cohortMethodData**: As created using `getDbCohortMethodData`
- **population**: The study population
- **prior**: object as created using `createPrior`
  - **priorType**: “laplace” or “none”
  - **variance**: variance of the prior (when not using cross-validation)
  - **useCrossValidation**: TRUE or FALSE
  - **exclude**: exclude these covariate IDs from regularization
- **control**: object as created using `createControl`
  - **tolerance**: numerical tolerance
  - **folds**: number of cross-validation folds
  - **cvRepetitions**: number of cross-validation repetitions
  - **threads**: number of CPU threads to use
createPs

Returns:

The study population data frame with an extra column for the propensity score
Diagnostics

- Did `createPs` complain about perfect prediction?
- Run `computePsAuc`: $0.5 < \text{AUC} < 1$?
- Run `getPsModel`: Strongest predictors are not the drugs of interest?
- Run `plotPs`: overlap between cohorts?

**Good:**

**Bad:**

![Good overlap between cohorts](image1.png)

![Bad overlap between cohorts](image2.png)
Now try it yourself!

• Run code in section *Propensity scores* up to `head(propensityModel)`
  – Modify the number of threads!

• Inspect the PS distribution plot

• Inspect the PS model

This is what I should get:
Step 4: Matching / Stratification / Trimming
Step 4: Matching / Stratification / Trimming

Matching
For every treated subject, select $n$ comparators using greedy matching

Stratification
Stratify into equally-sized strata based on PS

Trimming
Remove subjects with high and low PS
Arguments for both functions:

- **population**: population object with propensity scores
- **caliper**: maximum allowed difference in PS
- **caliperScale**: “standardized” or “propensity score”
- **maxRatio**: maximum number of comparators per target

Arguments for `matchOnPsAndCovariates`:

- **cohortMethodData**: As created using `getDbCohortMethodData`
- **covariateIds**: must match on these covariates
stratifyByPs & stratifyByPsAndCovariates

Arguments for both functions:

• **population**: population object with propensity scores
• **numberOfStrata**: number of strata

Arguments for stratifyByPsAndCovariates:

• **cohortMethodData**: As created using getDbCohortMethodData
• **covariateIds**: must match on these covariates
trimByPs & trimByPsToEquipoise

Argument for both functions:
• population

Argument for trimByPs:
• trimFraction: Fraction to be removed from each group

Argument for trimByPsToEquipoise:
• bounds: Bounds on the preference score
Diagnostics

- Run `getAttritionTable` or `drawAttritionDiagram`: did we not lose everyone?
- Run `computeCovariateBalance` and `plotCovariateBalanceScatterPlot`: standardized difference < 0.1 for all covariates?
Now try it yourself!

- Run the rest of the code in section *Propensity scores*
- Inspect the attrition diagram
- Inspect the balance scatter plot

This is what you should get:
Step 5: Fitting the outcome model
Step 5: Fitting the outcome model

Regression for outcome with at least treatment as predictor

Types:
- **Logistic**: compares risks
- **Poisson**: compares rates
- **Cox**: compares time-to-event

Conditioning:
- **Not conditioned**
- **Conditioned on matches set / strata**

Covariates:
- **None**
- **Same as used in propensity model**
fitOutcomeModel

Arguments:

• **population**: population with or without strata
• **cohortMethodData**: As created using `getDbCohortMethodData`
• **modelType**: “logistic”, “poisson”, or “cox”
• **stratified**: condition on strata?
• **useCovariates**: add same covariates as used in PS?
• **prior**: object as created using `createPrior`
  – **priorType**: “laplace” or “none”
  – **variance**: variance of the prior (when not using cross-validation)
  – **useCrossValidation**: TRUE or FALSE
  – **exclude**: exclude these covariate IDs from regularization
• **control**: object as created using `createControl`
  – **tolerance**: numerical tolerance
  – **folds**: number of cross-validation folds
  – **cvRepetitions**: number of cross-validation repetitions
  – **threads**: number of CPU threads to use
Diagnostics

• For Cox models run `plotKaplanMeier`: Evidence of non-proportionality?
Now try it yourself!

1. Run code in the section *Outcome Model*
2. Change the model to Cox regression

This is what you should get after step 2:

<table>
<thead>
<tr>
<th>Model type: cox</th>
<th>Stratified: TRUE</th>
<th>Use covariates: FALSE</th>
<th>Status: OK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>treatment</strong></td>
<td><strong>Estimate</strong></td>
<td><strong>lower .95</strong></td>
<td><strong>upper .95</strong></td>
</tr>
<tr>
<td></td>
<td>0.62500</td>
<td>0.18883</td>
<td>1.87314</td>
</tr>
</tbody>
</table>
Generic study steps

1. Getting the necessary data from the database
2. Defining the study population
3. [Creating a propensity model]
4. [Trimming / Matching / Stratification]
5. Fitting the outcome model

+ generating various diagnostics
All-by-all support

Drug – Comparator - Outcome

Analysis settings

CohortMethod

For:
- Sensitivity analyses
- Including negative controls
- Methods research
- Safety surveillance

Estimates, Diagnostics
Negative controls as study diagnostics

- Negative control outcomes are outcomes not believed to be caused by either exposure
- Assume true HR = 1
- Observe distribution of estimates
Negative control distribution

Approx. 95% of estimates should have 1 inside 95% confidence interval
Further things to try

• Change code to use all covariates (instead of handpicked ones)
• Remove people with prior outcomes
• Create a Kaplan-Meier plot
Counterfactual – other track

- Learned how to create the cohorts we used
Counterfactual – other track

- Learned how to create the cohorts we used
- Learned to think about the study design
- Learned how to use Atlas to generate starting R code