Causal Inference using Generalized Empirical Likelihood Methods with R

Pierre Chaussé* Mihai Giurcanu § George Luta **

November 9, 2020

Abstract

To be added

*Pierre Chaussé Email: pchausse@uwaterloo.ca (corresponding author). Department of Economics, University of Waterloo, Ontario, Canada

§Mihai Giurcanu Email: giurcanu@uchicago.edu. Department of Public Health Sciences, University of Chicago, 5841 S Maryland Ave, Room R325, Chicago, IL, 60637, USA

**George Luta Email: George.Luta@georgetown.edu. Department of Biostatistics, Bioinformatics and Biomathematics, Georgetown University

Contents

1 Introduction

In the following, let $X \in \mathbb{R}^q$ be a random q-vector of covariates, $Y(j)$ be the random (potential) outcome when the subject is exposed to the treatment j, where $0 \le j \le p$ and $j = 0$ corresponds to the control treatment. We consider Z_0 as an indicator for the control treatment, and in the case when the observational study does not have a well defined control treatment, then we treat Z_0 as the baseline treatment, i.e., the treatment with which we are interested to compare all other treatments. Let further $Z = (Z_j) \in \mathbb{R}^p$ be a random p-vector of treatment indicators (other than the control treatment), with $Z_j = 1$ and $Z_k = 0$ for all $k \neq j$ if the subject receives the treatment j, where $j = 1, \dots, p$. Since all individuals receive only one treatment, then $\sum_{j=0}^p Z_j = 1$, and we only observe $Y = \sum_{j=0}^p Y(j) Z_j$. Note that we do not consider here the case of clinical trials with non-compliance where the subjects are assigned to treatments but they may receive other treatments, and thus, the available data consist of only the treatment received and the outcome under the treatment received in addition to the covariates for all subjects.

1.1 Randomized experiments

Let $\theta_1 \in \mathbb{R}$, $\theta_2 = (\theta_{2,1}, \dots, \theta_{2,p}) \in \mathbb{R}^p$, and $\theta_3 = (\theta_{3,1}, \dots, \theta_{3,p}) \in \mathbb{R}^p$ be defined as $\theta_1 = \mathrm{E}(Y(0))$, $\theta_{2,j} = [\mathrm{E}(Y(j)) - \mathrm{E}(Y(0))]$, $\theta_{3,j} = \mathrm{E}(Z_j)$, $1 \le j \le p$, and let $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^{2p+1}$. By the definition of conditional expectation,

$$\mathrm{E}(Y|Z_j = 1) = \mathrm{E}(Y(j)) = \mathrm{E}(YZ_j)/\,\mathrm{E}(Z_j)\,, \quad 1 \le j \le p\,. \tag{1}$$

Hence, $\mathrm{E}[(Y - \theta_1 - \theta_2^T Z)Z_j] = 0$ for all $1 \le j \le p$, and thus,

$$\mathrm{E}\big((Y - \theta_1 - \theta_2^T Z)Z\big) = 0\,. \tag{2}$$

By the law of total expectation formula,

$$\mathrm{E}(Y) = \sum_{j=0}^{p} \mathrm{E}(Y(j))\,\mathrm{Pr}(Z_j = 1)\,. \tag{3}$$

Hence,

$$\mathrm{E}(Y - \theta_1 - \theta_2^T Z) = 0\,. \tag{4}$$

By the definition of θ_3, $\theta_{3,j} = \mathrm{E}(Z_j)$ for $1 \le j \le p$; hence

$$\mathrm{E}(Z - \theta_3) = 0\,. \tag{5}$$

Note that $\mathrm{E}(Z_0) = 1 - \sum_{j=1}^p \theta_{3,j}$.

In randomized trials, $(Z_0, Z) \perp\!\!\!\perp X$, which implies that $\mathrm{E}[(Z_j - \theta_{3,j})u(X)] = 0$ for $1 \le j \le p$, where $u(X) \in \mathbb{R}^k$ is a k-vector of functions of X. Hence,

$$\mathrm{E}[(Z - \theta_3) \otimes u(X)] = 0\,. \tag{6}$$

To illustrate what $u(x)$ can be, suppose $x = (x_1, x_2) \in \mathbb{R}^2$, then we can define $u(x) = x \in \mathbb{R}^2$, or $u(x) = (x_1, x_2, x_1 x_2) \in \mathbb{R}^3$, or $u(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2) \in \mathbb{R}^5$.

Let $T = (X, Z, Y) \in \mathbb{R}^{p+q+1}$ denote a generic random variable distributed according to a distribution on $\mathbb{R}^{p+q+1}$[1]. Therefore, Equations (2) and (4) to (6) imply that the parameter of interest θ^0 satisfies the following moment conditions:

$$\mathrm{E}(g(T, \theta^0)) = 0\,, \tag{7}$$

where $\theta^0 = (\theta_1^0, \theta_2^0, \theta_3^0) \in \mathbb{R}^{2p+1}$ and $g(t, \theta)$ is defined as

$$g(t;\theta) = \begin{pmatrix} y - \theta_1 - \theta_2^T z \\ (y - \theta_1 - \theta_2^T z)z \\ z - \theta_3 \\ (z - \theta_3) \otimes u(x) \end{pmatrix}\,, \quad t = (x, z, y) \in \mathbb{R}^{p+q+1}\,. \tag{8}$$

The ACE of the treatment j is given by $\tau_j^0 = \theta_{2,j}^0$, for $1 \leq j \leq p$.

The package offers a way to estimate θ^0 using the generalized method of moments (GEL). Using the primal form of GEL, the estimator of θ^0 is defined as:

$$\hat{\theta} = \mathrm{argmin}_{\theta \in \mathbb{R}^{2p+1}} \min_{p \in \mathbb{P}^n} \left\{ D_\gamma(p, n^{-1}1_n) : \sum_{i=1}^n p_i g(T_i; \theta) = 0 \right\}, \tag{9}$$

where

$$\mathbb{P}^n = \left\{ p = (p_i) \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1\,, p_i \geq 0 \right\},$$

and $D_\gamma(p, n^{-1}1_n)$ is the power divergence discrepancy function (Newey and Smith, 2004):

$$D_\gamma(p, n^{-1}1_n) = \sum_{i=1}^n \frac{(np_i)^{\gamma+1} - 1}{n\gamma(\gamma+1)}\,.$$

In particular, $\gamma = -1$ corresponds to the empirical likelihood (EL), $\gamma = 0$ corresponds to the exponential tilting (ET), $\gamma = 1$ corresponds to the Euclidean empirical likelihood (EEL) estimator also known as the continuously updated GMM estimator (CUE), and $\gamma = -1/2$ corresponds to the Hellinger distance (HD) used by Kitamura et al. (2013). Newey and Smith (2004) present the GEL method in its dual form, which is the following saddle point problem:

$$\hat{\theta} = \mathrm{argmin}_{\theta \in \mathbb{R}^{2p+1}} \max_{\lambda \in \mathbb{R}^{1+p(2+k)}} \sum_{i=1}^n \rho_\gamma\big(\lambda^T g(T_i; \theta)\big)\,, \tag{10}$$

where

$$\rho_\gamma(v) = -\frac{(1 + \gamma v)^{(\gamma+1)/\gamma}}{\gamma + 1}\,.$$

In particular $\rho_{-1}(v) = \log(1 - v)$ for EL, $\rho_0(v) = -\exp(v)$ for ET, $\rho_1(v) = -1/2 - v - v^2/2$ for EEL, and $\rho_{-1/2}(v) = -2/(1 - v/2)$ for HD. Using the dual form, the estimated probability weights from the primal problem are defined as:

$$\hat{p}_i(\theta, \lambda) = \frac{\rho'_\gamma(\lambda^T g(T_i; \theta))}{\sum_{j=1}^n \rho'_\gamma(\lambda^T g(T_i; \theta))}\,, \tag{11}$$

where $\rho'_\gamma(v)$ is the first order derivative of $\rho_\gamma(v)$.

[1] Notice that we omit Z_0 from T because its value is implied by Z through $Z_0 = 1 - \sum_{j=1}^p Z_j$.

1.2 Observational studies

When the treatment (group) assignment is not random, we can still use the GEL as a weighting method. GEL is used as a way to re-weight the probability of each observation so that our sample is as if it had been generated by a randomized experiment. The parameter of interest θ^0 satisfies the following moment conditions:

$$\mathrm{E}_0(g(T,\theta^0)) = 0\,, \tag{12}$$

where θ^0 is as in Section 1.1, and $g(t,\theta)$ is defined as

$$g(t;\theta) = \begin{pmatrix} y-\theta_1-\theta_2^T z \\ (y-\theta_1-\theta_2^T z)z \\ z-\theta_3 \\ (z-\theta_3)\otimes u(x) \\ u(x)-u_0 \end{pmatrix}\,, \quad t=(x,z,y)\in\mathbb{R}^{p+q+1}\,, \tag{13}$$

where u_0 is the expected value of $u(X)$ for a target population. Note that while the first three moment conditions (under E_0) identify the parameters, the fourth moment condition makes Z "almost independent" of X as $k\to\infty$. The last condition is what differentiates randomized experiments from observational studies. It imposes moments of X to match the ones from a given target population. The choice of u_0 is driven by the type of causal effect we are interested in. We will present the different options in the next section.

2 Estimating the causal effect

The package is based on the "momentfit" package (Chaussé, 2020), which offers ways to build classes for moment-based models, and algorithms to estimate them. It offers two ways to fit a model. We can either create the model object and call the *gmmFit* or **gelFit** method to estimate it by GMM or GEL, or diectly call the gmm4() or gel4() function. The "causalGEL" package is built in th same way. Section 2.1 presents the two-step way, which is useful is we want to fit the same model using different methods, and Section 3 presents the one-step way, using the causalGel() function.

2.1 An S4 class object for causal inference

To illustrate the methods, we consider the experiment analyzed first by Lalonde (1986) and used later by Dehejia and Wahba (1999, 2002). The objective of the original paper was to measure the effect of a training program on the real income. The dependent variable is the real income in 1978 and the covariates used for matching the treated group to the control are age, education, 1975 real income and dummy variables for race, marital status, and academic achievement.

First, we load the package and the dataset:

```
library(causalGel)
data(nsw)
## We express income in thousands for better stability
nsw$re78 <- nsw$re78/1000
nsw$re75 <- nsw$re75/1000
```

The model class, is "causalModel" which inherits directly from the "functionModel" class defined in the "modelfit" package. The constructor is the causalModel() function. The arguments are:

- *g*: A formula that defines the regression of the outome on the treatment indicator. For our dataset, the variable "treat" is the indicator, and "re78" is the outcome. The formula is therefore:

```
g <- re78~treat
```

- *balm*: A formula or a data.frame representing $u(X)$. For example, if we want to balance 1975 income, age, education and race, we would use the following:

```
balm <- ~age+ed+black+hisp+re75
```

- *theta0*: An optional starting value to be passed to the numerical algorithm.
- *momType*: This is the main argument to determine which type of causal effect we want to estimate. The options are:
  - "ACE": This one is for estimating the average causal effect. The moment function $g(t;\theta)$ is defined by Equation (13), and μ_0 is defined as:

$$\mu_0 = \frac{1}{n}\sum_{i=1}^{n} u(X_i)$$

  - "ACT": This is for the causal effect on the treated. In that case, the argument "ACTmom" determines which of the treated groups we are refering to. The moment function $g(t;\theta)$ is defined by Equation (13), and μ_0 is defined as:

$$\mu_0 = \frac{1}{n_j}\sum_{i=1}^{n} Z_{ji}u(X_i)\,,$$

    where $n_j = \sum_{i=1}^{n} Z_{ji}$, and j is the value of "ACTmom".
  - "ACC": This is for the causal effect on the control. The moment function $g(t;\theta)$ is defined by Equation (13), and μ_0 is defined as:

$$\mu_0 = \frac{1}{n_0}\sum_{i=1}^{n} Z_{0i}u(X_i)\,,$$

    where $n_0 = \sum_{i=1}^{n} Z_{0i}$.
  - "uncondBal": This is used to estimate the average causal effect in randomized trials. The moment function $g(t;\theta)$ is defined by Equation (8). In the case of observational data, it is not recommended because the moments are balanced, but represent estimates of the moments for an undefined population.
  - "fixedMom": The causal effect of a target population for which $E(\mu(X)$ is known. The moment function $g(t;\theta)$ is defined by Equation (13), and μ_0 is set to "popMom", which is another argument of causalModel() (see below).

- *popMom*: A $k \times 1$ vector, representing $\mathrm{E}(\mu(X))$. If provided, *momType* is automatically set to "popMom".
- *gelType*: The type of GEL method. The options are "EL" (the default), "ET", "EEL" and "HD", as defined above. The exponentially tilted empirical likelihood (ETEL) and exponentially tilted Hellinger distance (ETHD) are also available. The last available method is "REEL" which is the restricted EEL. The solution is obtained by restricting the EEL implied probability, defined in Equation (11), to be non-negative.
- *rhoFct*: An optional $\rho(v)$ function if the desired GEL method is not available in the package (see the GEL vignette from the "momentfit" package for more details).
- *data*: A data.frame with all the variables needed to evaluate the formulas g and *balm*.

The following are three different models:

```
ace <- causalModel(g, balm, nsw, momType="ACE")
act <- causalModel(g, balm, nsw, momType="ACT")
aceRT <- causalModel(g, balm, nsw, momType="uncondBal")
```

The third one is the ACE assuming randomized assignments. A print method for that class summarizes the model:

```
ace

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 13
## Number of balancing covariates: 5
## Sample size:  722
```

The option "printBalCov" allows us to see the balancing covariates:

```
print(act, printBalCov=TRUE)

## Causal Model
## *************
## Model type: Causal effect on the treated
## Number of treatments: 1
## Number of moment conditions: 13
## Number of balancing covariates: 5
## Sample size:  722
## Balancing covariates:
##   age, ed, black
##  hisp, re75
```

To add powers and interactions, we can follow the usual rules for formulas. Here is an example in which age is interacted with education, and 1975 income squared is included:

```
balm2 <- ~age*ed+black+hisp+re75+I(re75^2)
ace2 <- causalModel(g, balm2, nsw, momType="ACE")
print(ace2, printBalCov=TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 17
## Number of balancing covariates: 7
## Sample size:  722
## Balancing covariates:
##   age, ed, black
##  hisp, re75, I(re75^2)
##  age:ed
```

2.2 The *gelFit* method and the "causalGelfit" object

As mentioned in the previous section, the "causalModel" inherits from the "functionModel" class. The *gelFit* method is therefore a slightly modified method that calls the *gelFit* for "functionModel" objects, and creates a "causalGelfit" object. It inherits directly from "gelfit" class, but having a different one allows to build other methods such as *print* and *vcov*, that are specific to our model. The following *print* method offers the option of printing the $\hat{\lambda}$ and the model info.

```
fit1 <- gelFit(ace, gelType="EL") ## EL is the default
print(fit1, model=FALSE, lambda=TRUE)

##
## Estimation:  EL
## Convergence Theta:  0
## Convergence Lambda:  0
## coefficients:
##       control   causalEffect  probTreatment
##     5.0945926      0.8223392      0.4113576
## lambdas:
##       control   causalEffect  probTreatment      treat_age       treat_ed
## -5.171055e-07   1.432391e-06   5.289036e-01  -3.288704e-03  -5.962977e-02
##   treat_black     treat_hisp     treat_re75            age             ed
##  1.635445e-01   2.907282e-01   8.043465e-04  -5.643722e-05   4.119271e-03
##         black           hisp           re75
## -5.736834e-03   3.484319e-03  -8.729253e-04
```

The coefficients are labeled as "control" for θ_1, "causalEffect" for θ_2, and "probTreatment" for θ_3. For the case of multiple treatments, the treatment effect coefficients are labeled "causalEffecti", where i identifies the treatment. The following are the existing methods for "causalGelfit" objects. Since "causalGelfit" contains a "gelfit" object, most methods are the one built for "gelfit" objects. Here is a list:

- *vcov*: It computes the covariance matrix of $\hat{\theta}$ and $\hat{\lambda}$ in a list. The list contains other information used by other methods. We don't often need to run the method, but if needed, the covariance matrix of $\hat{\theta}$ is

```r
vcov(fit1)$vcov_par

##
##   7.481449e-02 -7.392353e-02 4.311762e-06
##  -7.392353e-02  2.242136e-01 2.762385e-05
##   4.311762e-06  2.762385e-05 3.353775e-04
```

By default, the covariance matrix is robust to misspecification, which is what we should used in observational studies. For randomized trials, we can set the argument "robToMiss" to FALSE, because it is not needed.

- *confint* The method computes a confidence interval. By default, it is a Wald type of confidence:

```r
confint(fit1)

##
##  Wald type confidence interval
##                  0.025   0.975
## control         4.5585  5.6307
## causalEffect   -0.1057  1.7504
## probTreatment   0.3755  0.4473
```

It is also possible to get an interval based on the inversion of the likelihoo ratio. The empirical likelihood confidence is:

```r
confint(fit1, 2, type="invLR")

##
##  Confidence interval based on the inversion of the LR test
##                  0.025  0.975
## causalEffect  -0.07155  1.808
```

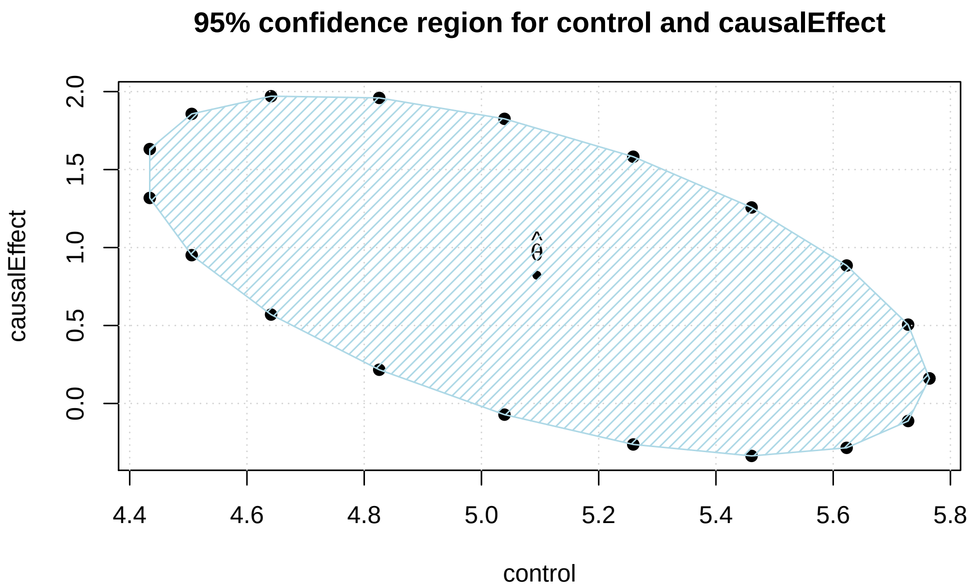
Confidence regions are also possible using a pair of coefficients:

```r
cr <- confint(fit1, 1:2, area=TRUE)
cr

## Wald type confidence region
## ***************************
## Level: 0.95
## Number of points: 20
## Variables:
##  Range for control: [4.434, 5.764]
##  Range for causalEffect: [-0.3362, 1.971]
```

This is an object of class "mconfint" for which a *plot* method exists:

```r
plot(cr, col="lightblue", density=20)
```



- *summary*: The method creates a "summaryGel" with its own *print* method. It returns an output similar to the summary method of "lm" objects.

```
summary(fit1)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 13
## Number of balancing covariates: 5
## Sample size:  722
##
## Estimation:  EL
## Convergence Theta: 0
## Convergence Lambda: 0
## Average |Sum of pt*gt()]|: 4.2944e-16
## |Sum of pt - 1|: 0
##
## coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## control       5.094593   0.273522 18.6259  < 2e-16 ***
## causalEffect  0.822339   0.473512  1.7367  0.08244 .
## probTreatment 0.411358   0.018313 22.4622  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Lambdas:
##                  Estimate  Std. Error  t value Pr(>|t|)
## control       -5.1711e-07  2.5083e-08 -20.6155   <2e-16 ***
## causalEffect   1.4324e-06  5.8736e-08  24.3871   <2e-16 ***
## probTreatment  5.2890e-01  6.0880e-01   0.8688   0.3850
## treat_age     -3.2887e-03  1.1630e-02  -0.2828   0.7773
```

```
## treat_ed      -5.9630e-02  4.4931e-02  -1.3271   0.1845
## treat_black    1.6354e-01  2.6588e-01   0.6151   0.5385
## treat_hisp     2.9073e-01  3.4583e-01   0.8407   0.4005
## treat_re75     8.0435e-04  1.5495e-02   0.0519   0.9586
## age           -5.6437e-05  7.1538e-04  -0.0789   0.9371
## ed             4.1193e-03  4.4968e-03   0.9160   0.3596
## black         -5.7368e-03  1.1078e-02  -0.5178   0.6046
## hisp           3.4843e-03  1.8667e-02   0.1867   0.8519
## re75          -8.7293e-04  9.3098e-04  -0.9376   0.3484
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Test E(g)=0
##       Statistics  df   pvalue
## LR:       3.0180  10  0.98100
## LM:       3.0169  10  0.98102
##  J:       3.0177  10  0.98100
```

2.3 Other useful methods

Instead of creating a new model with different balancing moments $\mu(X)$, it is possible to use the method "[" to subset the existing $\mu(X)$. We can think of a model object as being a two dimensional array, the first dimension being the balancing moments, and the second being the observations. Consider the following

```
ace <- causalModel(re78~treat,
                   ~(age+black+ed)*(age+black+ed) + I(age^2) + I(ed^2),
                   data=nsw)
print(ace, TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 19
## Number of balancing covariates: 8
## Sample size:  722
## Balancing covariates:
##   age, black, ed
##  I(age^2), I(ed^2), age:black
##  age:ed, black:ed
```

Suppose we want to removed the squared components:

```
print(ace[-c(3,4)], TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 15
## Number of balancing covariates: 6
```

```
## Sample size:  722
## Balancing covariates:
##   age, black, I(ed^2)
##  age:black, age:ed, black:ed
```

Or remove interactions

```
print(ace[1:5], TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 13
## Number of balancing covariates: 5
## Sample size:  722
## Balancing covariates:
##   age, black, ed
##  I(age^2), I(ed^2)
```

We can use a subset of the sample by adding a second argument:

```
print(ace[,1:100], TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 19
## Number of balancing covariates: 8
## Sample size:  100
## Balancing covariates:
##   age, black, ed
##  I(age^2), I(ed^2), age:black
##  age:ed, black:ed

print(ace[1:3,nsw$re75>0], TRUE)

## Causal Model
## *************
## Model type: Average causal effect
## Number of treatments: 1
## Number of moment conditions: 9
## Number of balancing covariates: 3
## Sample size:  433
## Balancing covariates:
##   age, black, ed
```

An easy way to re-estimate a new model specified by "[", is to use the method for "causalGelfit" objects. It changes the model and re-estimate it.

	Model 1	Model 2	Model 3
control	5.1263***	5.1627***	5.1013***
	(0.2795)	(0.2972)	(0.2777)
causalEffect	0.7713	0.6855	0.4535
	(0.4798)	(0.4876)	(0.7817)
probTreatment	0.4114***	0.4120***	0.1500***
	(0.0183)	(0.0184)	(0.0160)
Num. obs.	722	716	500
Num. Bal. Cov.	8	8	3

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: Statistical models

```
fit <- gelFit(ace)
fit2 <- fit[,nsw$age<48]
fit3 <- fit[1:3,1:500]
```

The results are shown in Table 1, which is constructed using the "texreg" package of Leifeld (2013). The code for the *extract* method is in the appendix.

Details about the convergence are obtained using the *checkConv* method:

```
checkConv(fit)

## Convergence details of the Causal estimation
## ********************************************
## Average causal effect
##
## Convergence of the Lambdas: TRUE
## Convergence of the Coefficients: TRUE
## Achieved moment balancing: TRUE
##
## Moments for each group:
##               treat=0    treat=1
## age         24.520776  24.520776
## black        0.800554   0.800554
## ed          10.267313  10.267313
## I(age^2)   645.110803 645.110803
## I(ed^2)    108.319945 108.319945
## age:black   19.876731  19.876731
## age:ed     252.038781 252.038781
## black:ed     8.265928   8.265928
```

It compares sample moments of $\mu(X)$ for each group, using the estimated implied probabilities. We can then see if the balancing was achieved. As an example, the first column is $[\sum_{i=1}^n \hat{p}_i(1-Z_i)\mu(X_i)]/[\sum_{i=1}^n \hat{p}_i(1-Z_i)]$, and the second column is $[\sum_{i=1}^n \hat{p}_i Z_i\mu(X_i)]/[\sum_{i=1}^n \hat{p}_i Z_i]$, which are respectively estimates of $\mathrm{E}(\mu(X)|Z=0)$ and $\mathrm{E}(\mu(X)|Z=1)$. We can see that the moments are well balanced, at least up to six decimals.

	ACE(rand.)	ACE(non-random)	ACT	ACC
control	5.0969*	5.0946*	5.1047*	5.0901*
	[4.5606; 5.6332]	[4.5585; 5.6307]	[4.5450; 5.6644]	[4.5471; 5.6331]
causalEffect	0.8157	0.8223	0.8717	0.7892
	[−0.1108; 1.7423]	[−0.1057; 1.7504]	[−0.0838; 1.8272]	[−0.1325; 1.7108]
probTreatment	0.4114*	0.4114*	0.4114*	0.4114*
	[0.3755; 0.4473]	[0.3755; 0.4473]	[0.3755; 0.4473]	[0.3755; 0.4473]
Num. obs.	722	722	722	722
Num. Bal. Cov.	5	5	5	5

* Null hypothesis value outside the confidence interval.

Table 2: Causal Effect for a Training Program

3 The causalGEL function

The function allows to estimate the causal effect without having to go through the step of creating the model. The different arguments are a mixture of the arguments of causalModel(), the *solveGel* method of the "momentfit" package, and the *gelFit* method. The average causal effect of the training program, assuming random assignment and using EL, can be obtained as follows:

```
data(nsw)
nsw$re78 <- nsw$re78/1000
nsw$re75 <- nsw$re75/1000
fit1 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                 momType="uncondBal")
```

Similarly, the ACE, ACT and ACC can be computed as follows (The results are presented in Table 2).

```
fit2 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="ACE")
fit3 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="ACT")
fit4 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="ACC")
```

It is also possible to estimate restricted models, by passing restrictions to the arguments "cstLHS" and "cstRHS". There are two possible approaches. The first one is to define the restrictions in a vector of characters. In that case, "cstRHS" is set to its default value. For example, if we want to restrict the causal effect coefficient to be equal to 1, we proceed as:

```
causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
          momType="uncondBal", cstLHS="causalEffect=1")

## Causal Model
## *************
## Model type: Unconditional balancing
## Number of treatments: 1
```

```
## Number of moment conditions: 8
## Number of balancing covariates: 5
## Sample size:  722
## Additional Specifications: Restricted model
## Constraints:
##  causalEffect ~ 1
##
## Estimation:  EL
## Convergence Theta:  0
## Convergence Lambda:  0
## coefficients:
##       control  probTreatment
##     5.0384965      0.4113554
```

If we want the above restriction plus the probability of being in the treatment group to be equal 0.5, we proceed this way. Notice that the restricted model only has one coefficient. To avoid complains coming from optim(), which warns you that Nelder-Mead is not reliable in one-dimensional optimization problems, we set the method to "Brent" using the "tControl" argument:

```
causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
          momType="uncondBal", cstLHS=c("causalEffect=1", "probTreatment=0.5"),
          tControl=list(method="Brent", lower=0, upper=10))

## Causal Model
## *************
## Model type: Unconditional balancing
## Number of treatments: 1
## Number of moment conditions: 8
## Number of balancing covariates: 5
## Sample size:  722
## Additional Specifications: Restricted model
## Constraints:
##  causalEffect ~ 1
##  probTreatment ~ 0.5
##
## Estimation:  EL
## Convergence Theta:  0
## Convergence Lambda:  0
## coefficients:
##  control
## 5.038502
```

The problem with the above approach is that we need to know the names of the coefficients before calling causalGel(). For equallity constraints, we can instead set "cstLHS" to the coefficient positions, and "cstRHS" to their restricted values. The above two restricted models can therefore be obtained as follows:

```
causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
          momType="uncondBal", cstLHS=2, cstRHS=1)@theta

##       control probTreatment
##     5.0384965     0.4113554
```

```
causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
          momType="uncondBal", cstLHS=2:3, cstRHS=c(1,.5),
          tControl=list(method="Brent", lower=0, upper=10))@theta

##  control
## 5.038502
```

Notice that it is also possible to create a restricted model and follow the method described in Section 2. To create the above two restricted models, we first create the unrestricted one:

```
Un_model <- causalModel(re78~treat, ~age+ed+black+hisp+re75, nsw,
                    momType="uncondBal")
```

Then, we use the *restModel* method from the "momentfit" package:

```
restModel(Un_model, causalEffect~1)

## Causal Model
## *************
## Model type: Unconditional balancing
## Number of treatments: 1
## Number of moment conditions: 8
## Number of balancing covariates: 5
## Sample size:  722
## Additional Specifications: Restricted model
## Constraints:
##  causalEffect ~ 1

# or restModel(Un_model, "causalEffect=1")
restModel(Un_model, list(causalEffect~1, probTreatment~0.5))

## Causal Model
## *************
## Model type: Unconditional balancing
## Number of treatments: 1
## Number of moment conditions: 8
## Number of balancing covariates: 5
## Sample size:  722
## Additional Specifications: Restricted model
## Constraints:
##  causalEffect ~ 1
##  probTreatment ~ 0.5

# or restModel(Un_model, c("causalEffect=1", "probTreatment=0.5"))
```

The *gelFit* method can then be applied to the restricted models.

3.1 Restricting the λ's

For the moment conditions defined by Equations 13 and 8, the analytical solution of the λ's associated with the first two lines is exactly 0. However, the numerical solution is not exactly zero:

```
fit1 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="uncondBal")
fit1@lambda[1:2]

##      control causalEffect
## 3.924937e-08 7.426004e-07
```

It may be faster and more precise to restrict these λ's to be zero. If we set the option reatrictLam to TRUE, these lambda are fixed at 0 and the coefficients associated with the causal effect equation are computed by solving:

$$\sum_{i=1}^{n} \hat{p}_i(\hat{\theta}, \hat{\lambda}) \begin{pmatrix} Y_i - \theta_1 - \theta_2^T Z_i \\ (Y_i - \theta_1 - \theta_2^T Z_i) Z_i \end{pmatrix} .$$

Notice that the implied probabilities do not depend on θ_1 and θ_2, which is why we can get them first and then solve for θ_1 and θ_2. We can see that the results are similar, but it may speedup estimation especially in simulation studies.

```
fit2 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="uncondBal", restrictLam=TRUE)
rbind(coef(fit1), coef(fit2))

##       control causalEffect probTreatment
## [1,] 5.096869    0.8157308     0.4113575
## [2,] 5.096868    0.8156973     0.4113550

rbind(fit1@lambda, fit2@lambda)

##           control causalEffect probTreatment    treat_age    treat_ed
## [1,] 3.924937e-08 7.426004e-07     0.5127469 -0.003384372 -0.05793918
## [2,] 0.000000e+00 0.000000e+00     0.5127335 -0.003384329 -0.05793893
##      treat_black treat_hisp   treat_re75
## [1,]   0.1635749  0.2928064 0.0008990034
## [2,]   0.1635743  0.2928069 0.0008990083
```

3.2 Using orthogonal bases

When the number of balancing moments increases, it may become numerically unstable to use them directly. It is also likely that they become collinear. To avoid the problem, we can replace the balancing matrix by the matrix of orthogonal bases that span the same space. We borrowed the function orth() from the pracma package (Borchers, 2019). This is done by adding the option orthoBases=TRUE. The following is used to compare the results, which are shown in Table 3. We can see that in most cases, there is very little difference.

```
fit1 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="ACT")
fit2 <- causalGEL(re78~treat, ~age+ed+black+hisp+re75, nsw, gelType="EL",
                  momType="ACT", orthoBases=TRUE)
```

	Original	Orthogonal Bases
control	5.104714***	5.104703***
	(0.285569)	(0.285568)
causalEffect	0.871684	0.871709
	(0.487502)	(0.487503)
probTreatment	0.411359***	0.411357***
	(0.018313)	(0.018313)
Num. obs.	722	722
Num. Bal. Cov.	5	5

***$p < 0.001$; **$p < 0.01$; *$p < 0.05$

Table 3: Comparing estimates with and without the orthogonal bases

The difference can be seen by looking at the ouput from the checkConv method:

```
checkConv(fit2)

## Convergence details of the Causal estimation
## *******************************************
## Causal effect on the treated
##
## Convergence of the Lambdas: TRUE
## Convergence of the Coefficients: TRUE
## Achieved moment balancing: TRUE
##
## Moments for each group:
##              treat=0       treat=1
## Basis1 -0.0364384452 -0.0364384452
## Basis2  0.0001946842  0.0001946842
## Basis3  0.0074397641  0.0074397641
## Basis4  0.0002770831  0.0002770831
## Basis5  0.0001206859  0.0001206859
```

The method no longer compare the moments of the original data (age, educ, black, etc.) but the moments of the bases. If some moments happen to be nearly collinear, we may see fewer bases.

References

Hans W. Borchers. *pracma: Practical Numerical Math Functions*, 2019. URL `https://CRAN.R-project.org/package=pracma`. R package version 2.2.9.

P. Chaussé. *momentfit: Methods of moments*, 2020. URL `https://r-forge.r-project.org/projects/gmm/`. R package version 0.1-0.

R.H. Dehejia and S. Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association*, 94(448), 1999.

R.H. Dehejia and S. Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and Statistics*, 84(1):151–161, 2002.

Y. Kitamura, T. Otsu, and K. Evdokimov. Robustness, infinitesimal neighborhoods, and moment restrictions. *Econometrica*, 81(3):1185–1201, 2013.

R. Lalonde. Evaluating the econometric evaluations of training programs. *American Economic Review*, 76:604–620, 1986.

Philip Leifeld. texreg: Conversion of statistical model output in R to LaTeX and HTML tables. *Journal of Statistical Software*, 55(8):1–24, 2013. URL `http://www.jstatsoft.org/v55/i08/`.

W. K. Newey and R. J. Smith. Higher order properties of gmm and generalized empirical likelihood estimators. *Econometrica*, 72:219–255, 2004.

A Some extra codes

The following *extract* is used with the "texreg" package of Leifeld (2013) to produce nice latex tables.

```
library(causalGel)
library(texreg)
setMethod("extract", "causalGelfit",
          function(model, includeSpecTest=FALSE,
                   specTest=c("LR","LM","J"), include.nobs=TRUE,
                   include.obj.fcn=TRUE, ...)
              {
                  specTest <- match.arg(specTest)
                  s <- summary(model, ...)
                  wspecTest <- grep(specTest, rownames(s@specTest@test))
                  spec <- modelDims(model@model)
                  coefs <- s@coef
                  names <- rownames(coefs)
                  coef <- coefs[, 1]
                  se <- coefs[, 2]
                  pval <- coefs[, 4]
                  n <- model@model@n
                  gof <- numeric()
                  gof.names <- character()
                  gof.decimal <- logical()
                  if (includeSpecTest) {
                      if (spec$k == spec$q)
                          {
                              obj.fcn <- NA
                              obj.pv <- NA
                          } else {
                              obj.fcn <- s@specTest@test[wspecTest,1]
                              obj.pv <- s@specTest@test[wspecTest,3]
                          }
                      gof <- c(gof, obj.fcn, obj.pv)
                      gof.names <- c(gof.names,
```

```
                                  paste(specTest,"-test Statistics", sep=""),
                                  paste(specTest,"-test p-value", sep=""))
                gof.decimal <- c(gof.decimal, TRUE, TRUE)
            }
            if (include.nobs == TRUE) {
                gof <- c(gof, n)
                gof.names <- c(gof.names, "Num.\\ obs.")
                gof.decimal <- c(gof.decimal, FALSE)
            }
            nbal <- length(model@model@X@balCov)
            gof.names <- c(gof.names, "Num. Bal. Cov.")
            gof <- c(gof, nbal)
            gof.decimal <- c(gof.decimal, FALSE)
            tr <- createTexreg(coef.names = names, coef = coef,
                               se = se, pvalues = pval,
                               gof.names = gof.names, gof = gof,
                               gof.decimal = gof.decimal)
            return(tr)
        })
```