SimulationComparisonWithCompetitors

Disclaimer: These simulations are not necessarily representative of the performance of these methods in the real-world. The simulation models are all randomly generated main-term parametric models and there are little-to-no positivity issues. Because of this, glm/glmnet will do better than other machine-learning algorithms in the below simulations. To make the simulations more difficult, take a look at the arguments to the function sim.causalGLM. The main point of these simulations is show the robust performance of causalGLM in small sample sizes and settings where parametric glm would do well. We hope these simulations convince you that semiparametric methods need not come at a cost in power/robustness relative to glm in regimes with small sample sizes, even when the truth is a simple parametric model.

Random examples of simulation datasets

```
library(causalGLM)
```

```
## Loading required package: sl3

## Loading required package: hal9001

## Loading required package: Rcpp

## hal9001 v0.4.0: The Scalable Highly Adaptive Lasso
## note: fit_hal defaults have changed. See ?fit_hal for details

## Loading required package: data.table

## Loading required package: R6
```

```
n <- 50
p <- 5
data_sim <- sim.CATE(n=n, p=p, formula_estimand =  ~1 + W1 + W2 + W3 + W4, formula_A = ~ ., formula_YOW
# simulated data and true nuisance functions
head(data_sim$data)
```

```
##           W1         W2          W3          W4          W5 A          Y
## 1 -0.8572854 -0.1390749 -0.85049042  0.09343069  0.07784028 0 -0.3521148
## 2  0.0148692  0.8337207  0.59934040  0.37717416  0.24728231 0 -0.3600709
## 3 -0.3129057  0.8566992  0.07401282 -0.37473451  0.77714590 0  0.1424504
## 4  0.6364820  0.2399171  0.43536470 -0.65268471 -0.68618412 0 -0.2982865
## 5  0.5638772  0.5767302 -0.22648430  0.04481764  0.69670380 0  0.5359837
## 6  0.9601558 -0.5622742  0.37961431  0.67401025 -0.48551372 0 -0.7597186
##         pA1         pY        sd       CATE
## 1 0.3730456 -0.6034048 0.1905959 -0.4547634
## 2 0.4107891 -0.2167146 0.1905959  1.1784102
## 3 0.3210399  0.2328711 0.1905959  0.4203839
## 4 0.3559859 -0.1744655 0.1905959  0.3560994
## 5 0.3113777  0.3132277 0.1905959  0.4835433
## 6 0.4602974 -0.7308713 0.1905959  0.6984970
```

```
# True coefs
data_sim$beta_CATE
```

```
## [1] 0.2176242 0.1749897 0.4818654 0.5938147 0.5317060
```

```
n <- 50
p <- 5
data_sim <- sim.RR(n=n, p=p, formula_estimand =  ~1 + W1 + W2 + W3 + W4, formula_A = ~ ., formula_YOW =
# simulated data and true nuisance functions
head(data_sim$data)
```

```
##             W1         W2          W3          W4         W5 A Y       pA1
## 1 -0.21321716 -0.2593888  0.19621651 -0.94967899  0.1512471 0 2 0.3709678
## 2  0.05099678  0.4553587  0.68045727 -0.32278016 -0.6850120 1 1 0.4242058
## 3  0.25439207 -0.3693566 -0.15549955 -0.05172519 -0.8340864 0 0 0.4525898
## 4  0.44188656 -0.4429626 -0.07146292 -0.02205594 -0.6020617 0 1 0.4558203
## 5 -0.22289668 -0.8314884  0.29871234  0.78127478  0.8371775 0 2 0.4629063
## 6  0.77029892 -0.2517619 -0.34654055 -0.81186595  0.8895011 0 1 0.3179203
##         pY        RR
## 1 2.725480 1.1709042
## 2 1.053798 1.2956859
## 3 1.678712 1.1701153
## 4 1.723518 1.1628322
## 5 1.396861 0.8774375
## 6 2.634059 1.3928992
```

```
# True coefs
data_sim$beta_logRR
```

```
## [1]  0.1931666  0.1241302  0.2413340 -0.1117586 -0.0796107
```

```
n <- 50
p <- 5
data_sim <- sim.OR(n=n, p=p, formula_estimand =  ~1 + W1 + W2 + W3 + W4, formula_A = ~ ., formula_YOW =
# simulated data and true nuisance functions
head(data_sim$data)
```

```
##            W1          W2         W3          W4         W5 A Y       pA1
## 1  0.7420075  0.16853068  0.8637087  0.59542990  0.2053032 1 0 0.3856068
## 2  0.6408653  0.82736436 -0.6629053 -0.22710439  0.2596439 1 0 0.4909074
## 3 -0.4214575  0.33158781  0.1468384 -0.66057934  0.7519537 1 1 0.5079269
## 4  0.3136387 -0.01029437  0.6937883  0.03582184 -0.8428803 0 1 0.3812178
## 5  0.2718144 -0.96770050 -0.6831317 -0.28343150 -0.4500423 0 1 0.4562895
## 6  0.2452843  0.68042639  0.5436341 -0.92355941  0.3666104 0 1 0.4206938
##          pY       pY1       pY0        OR
## 1 0.1912525 0.1912525 0.3278371 0.4848535
## 2 0.2105094 0.2105094 0.2469887 0.8129217
## 3 0.5917059 0.5917059 0.4852127 1.5375475
## 4 0.4940845 0.4165316 0.4940845 0.7309833
## 5 0.5466450 0.3511793 0.5466450 0.4488872
## 6 0.4085226 0.5594500 0.4085226 1.8386034
```

```
# True coefs
data_sim$beta_logOR
```

```
## [1] -0.3323576 -0.3509308  0.4214016  0.2267897 -0.6685204
```

Simulation Comparison with estimating-equation competitors

The sim.R function contains customizable functions that randomly generate test data for both the CATE, RR, and OR functions. The functions sim.causalGLM and sim.causalGLMwithLasso internally call these functions and run nsims number of simulations and report the proportion of estimated 95% confidence intervals that contain the true coefficient values (as determined from the simulation). The confidence interval coverage for the competitor, estimating equation-based estimators, is also reported. Both methods are fit on the same simulation data with the same nuisance estimators and same variance estimator. Therefore, the randomness is only in the difference between the two methods. We will employ these methods to compare the TMLE method implemented in this package, causalGLM, with competitors.

We will focus on small n with a 4-dimensional covariate model for the estimand. In these settings, differences are especially pronounced. The simulation data distributions are linear main-term parametric models, so that glm and glmnet are correctly specified. This may be unrealistic in some settings but it is an important benchmark. We would like these methods to do just as well as parametric glm in coverage when the assumptions are true.

Note sometimes due to montecarlo randomness one method may by chance perform better than another. For most reliable and fair results, set “nsims” as high as possible, e.g. nsims = 1000 or 2500.

Note the competitor is asyptotically equivalent to causalGLM. There for n large enough (e.g. n=1000), any differences should become smaller.

```
library(causalGLM)
seed <- 12345

nsims <- 200
```

CATE

The estimating equation for CATE is fairly simple (it is linear). Thus, estimating equation estimators should perform better for the CATE relative to non-linear estimating equation estimators (like for RR). ### n = 50, p=5

```
set.seed(seed)
n <- 50
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)          W1          W2          W3          W4 
##       1.000       0.965       0.970       0.995       0.985 
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.850 0.865 0.875 0.880 0.855
```

```
### You should see causalGLM do much better than the estimating equation estimator. The >0.95 coverage
```

n = 100, p=5

```
set.seed(seed)
n <- 100
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)           W1           W2           W3           W4
##           1            1            1            1            1
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.930 0.895 0.885 0.925 0.915
```

```
### You should see causalGLM do better than the estimating equation estimator. The >0.95 coverage is li
```

n = 200, p=5

```
set.seed(seed)
n <- 200
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)           W1           W2           W3           W4
##       1.000        0.990        0.995        0.990        0.980
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.945 0.915 0.955 0.945 0.900
```

```
### You should see both methods do fairly well. But, causalGLM does a bit better than the estimating eq
```

RR

The estimating equation for RR is highly non-linear. This should lead to worse performance for the estimating equation in certain settings. ### n = 50, p=5

```
set.seed(seed)
n <- 50
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These t
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)          W1          W2          W3          W4
##       0.985       0.990       0.990       0.980       1.000
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.84 0.85 0.82 0.83 0.81
```

```
### You should see causalGLM do much better than the estimating equation estimator. The >0.95 coverage
```

n = 100, p=5

```
set.seed(seed)
n <- 100
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "g
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These t
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)          W1          W2          W3          W4
##       0.975       0.990       0.995       0.990       0.990
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.835 0.865 0.870 0.895 0.855
```

```
### You should see causalGLM do better than the estimating equation estimator. The >0.95 coverage is li
```

n = 200, p=5

```
set.seed(seed)
n <- 200
p <- 5
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "g
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These t
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)          W1          W2          W3          W4
##       0.985       0.995       0.960       0.985       0.980
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.940 0.925 0.900 0.935 0.930
```

```
### You should see causalGLM do better than the competing estimating equation method.
```

OR

n = 50, p=5

```
set.seed(seed)
n <-   50
p <- 4
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
  out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 +W2 + W3 + W4   , n=n, p = p, learning_method =
```

```
## Warning in spOR(formula_logOR = formula, W, A, Y, pool_A_when_training =
## pool_A_when_training, : Targeting did not converge.

## Warning in spOR(formula_logOR = formula, W, A, Y, pool_A_when_training =
## pool_A_when_training, : Targeting did not converge.
```

```
# Both do great here!

x <-  sim.OR(formula_estimand = ~1 + W1 + W2 + W3 + W4, n=n, p=p)
x$data[, -c(1:5)]
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## [1] 0.940 0.965 0.960 0.960 0.955
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.935 0.975 0.985 0.970 0.975
```

```
# Both methods do great here!
```

n = 100, p=5

```
set.seed(seed)
n <- 100
p <- 4
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## [1] 0.955 0.940 0.950 0.955 0.950
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.945 0.955 0.960 0.955 0.970
```

```
### Similar performance again
```

n = 200, p=5

```
set.seed(seed)
n <- 200
p <- 4
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.
 out <-  sim.causalGLM(cross_fit = F, formula = ~1 + W1 + W2 + W3 + W4, n=n, p = p, learning_method = "
```

```
# The report function summarizes the coverage
# The first row of values is the coverage probability for each coeficient obtained by causalGLM. These
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## [1] 0.935 0.910 0.905 0.960 0.905
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.935 0.910 0.910 0.960 0.910
```

```
### You should see causalGLM do better than the competing estimating equation method.
```

High dimensional comparison with causalGLMwithLASSO

```
nsims <- 50
```

```
set.seed(seed)
n <- 50
p <- 200
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.

bigformula <- ~ 1 + W12 + + W2 + W3 + W112 + W123 + W154 + W45 + W34 + W121  + W63 + W164 + W162 + W63

smallformula <- ~ 1 + W12 + W50  + W99 + W14

out <-   sim.causalGLMwithLasso(formula = smallformula, n = n, p = p, formula_A = bigformula, formula_Y
```

```
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)         W12         W50         W99         W14
##        1.00        0.96        0.94        0.98        0.94
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.82 0.78 0.76 0.68 0.86
```

```
### You should see causalGLM do substantially better than the competing estimating equation method.
```

```
set.seed(seed)
n <- 100
p <- 200
#n=50 is sample size
#p=4 is number of covariates in W
```

```r
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.

bigformula <- ~ 1 + W12 + + W2 + W3 + W112 + W123 + W154 + W45 + W34 + W121  + W63 + W164 + W162 + W63 

smallformula <- ~ 1 + W12 + W50  + W99 + W14

out <-   sim.causalGLMwithLasso(formula = smallformula, n = n, p = p, formula_A = bigformula, formula_Y
```

```r
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)         W12         W50         W99         W14 
##        1.00        0.96        0.96        0.92        0.92 
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.90 0.92 0.90 0.86 0.90
```

```r
### You should see causalGLM do substantially better than the competing estimating equation method.
```

```r
set.seed(seed)
n <- 250
p <- 200
#n=50 is sample size
#p=4 is number of covariates in W
### This will print updates of coverage per iteration to your consol. It should run in a minute or so.

bigformula <- ~ 1 + W12 + + W2 + W3 + W112 + W123 + W154 + W45 + W34 + W121  + W63 + W164 + W162 + W63 

smallformula <- ~ 1 + W12 + W50  + W99 + W14
# Cross-fitting is turned off for speed
out <-   sim.causalGLMwithLasso(formula = smallformula, n = n, p = p, formula_A = bigformula, formula_Y
```

```r
 out$report()
```

```
## [1] "Coverage probability of 95% confidence intervals of causalGLM so far: "
## (Intercept)         W12         W50         W99         W14 
##        1.00        0.94        0.96        0.98        0.96 
## [1] "Coverage probability of 95% confidence intervals of the estimating equation (DML) competitor so
## [1] 0.90 0.90 0.96 0.96 0.94
```

```r
### The two methods perform similar, but it looks like causalGLM has a slight edge.
```