An Example of `plspolychaos` Use: ”plant”

J.P. Gauchi and A. Bouvier

MaIAGE, INRA, Université Paris-Saclay,
78350 Jouy-en-Josas,
France

May 4, 2016

Abstract

The **plspolychaos** R package computes sensitivity indexes from polynomial chaos expansions and regression PLS, for computer models with correlated continuous inputs. The functionalities and method are explained in the in-line manual of the package.

This paper illustrates the way of using the package on an example dataset, named `plant`, which has 9800 rows and 11 correlated inputs. The dataset is stored in the file `plant.Rda` in the directory `extdata` of the delivered package. We analyze the full polynomial of degree 5, which corresponds to 4387 monomials.

Note that another example is available in the vignette `ishigami20000` of the package.

Contents

1 Read Data

The data-frame `plant` is loaded. The 11 first columns are the input values. The following one is the response (output). Only, the 9800 first rows are taken into account. The function **descrdata** displays the main characteristics of the dataset.

```
> library("plspolychaos")
> load(system.file("extdata", "plant.Rda", package="plspolychaos"))
> X <- plant[1:9800, 1:11] #inputs
> Y <- plant[1:9800, 12] #response
> colnames(X) <- c("lati", "day", "xmax", "ymax","voxel", "height",
      "width", "LA", "k", "RH50","b")
> descrdata(X, Y) # data main characteristics
```

```
Number of rows: 9800

               Mean      Std Dev      Minimum      Maximum
lati   7.217307e-02 3.849487e+01 -66.00000000    66.000000
day    1.831420e+02 1.051579e+02   1.00000000   365.000000
xmax   2.496743e+00 8.678849e-01   1.00026122     4.000000
ymax   2.508623e+00 8.620006e-01   1.00022624     3.999859
voxel  1.289306e+01 4.751031e+00   1.00000000    20.000000
height 1.174070e+02 6.613806e+01   1.56786200   249.986355
width  1.017852e+02 5.394835e+01   1.07176129   200.000000
LA     5.019870e+04 2.803214e+04  18.26823614 99996.493840
k      5.750954e-01 3.034576e-01   0.01013429     1.099980
RH50   5.091690e-01 2.853858e-01   0.01020006     1.000000
b      3.026620e+00 1.710778e+00   0.01019729     5.999179
Y      1.822038e+01 1.780413e+01   2.00098700    97.927391

Correlation
               lati           day          xmax         ymax        voxel
lati    1.000000000 -0.0040311367 -0.0126461792 -0.019251730  0.005352445
day    -0.004031137  1.0000000000 -0.0087118307  0.003030392  0.004679213
xmax   -0.012646179 -0.0087118307  1.0000000000 -0.006372962  0.007650261
ymax   -0.019251730  0.0030303917 -0.0063729621  1.000000000 -0.011133154
voxel   0.005352445  0.0046792134  0.0076502611 -0.011133154  1.000000000
height  0.004399060  0.0001728457  0.0036641811  0.010900922  0.352589923
width   0.006696692 -0.0082791272  0.0005231269 -0.001815193  0.165663185
LA      0.004967966  0.0054488559 -0.0047471662 -0.007507580  0.114768601
k       0.002392601 -0.0079241772  0.0024449796 -0.005087792 -0.007312494
RH50    0.008750658 -0.0191320235  0.0129943451 -0.004991631  0.066883180
b       0.016166996 -0.0045653446 -0.0038672159 -0.005128775  0.088979019
Y      -0.005557504 -0.0018613193  0.0035338251  0.004020859  0.073942068
              height         width           LA            k         RH50
lati    0.0043990602  0.0066966918  0.004967966  0.002392601  0.008750658
day     0.0001728457 -0.0082791272  0.005448856 -0.007924177 -0.019132024
xmax    0.0036641811  0.0005231269 -0.004747166  0.002444980  0.012994345
ymax    0.0109009215 -0.0018151932 -0.007507580 -0.005087792 -0.004991631
voxel   0.3525899230  0.1656631848  0.114768601 -0.007312494  0.066883180
height  1.0000000000  0.5189667457  0.234968526 -0.113601020  0.182468490
width   0.5189667457  1.0000000000  0.276403531  0.136192150  0.172670233
LA      0.2349685263  0.2764035314  1.000000000 -0.080260961 -0.003031810
k      -0.1136010205  0.1361921502 -0.080260961  1.000000000  0.213624164
```

```
RH50    0.1824684897  0.1726702326 -0.003031810  0.213624164  1.000000000
b       0.2380747394  0.2166877215  0.011580900 -0.055415774  0.081545633
Y      -0.5564700966 -0.6106980024 -0.022873441  0.142834896 -0.081923803
                 b            Y
lati    0.016166996 -0.005557504
day    -0.004565345 -0.001861319
xmax   -0.003867216  0.003533825
ymax   -0.005128775  0.004020859
voxel   0.088979019  0.073942068
height  0.238074739 -0.556470097
width   0.216687722 -0.610698002
LA      0.011580900 -0.022873441
k      -0.055415774  0.142834896
RH50    0.081545633 -0.081923803
b       1.000000000 -0.222542127
Y      -0.222542127  1.000000000
```

2 Building Legendre Polynomials

We build Legendre polynomials of degree 5 by using the function **polyLeg**. An object of class PCEpoly is created. Its method **print** displays its main characteristics.

```
> pcet <- polyLeg(X, Y, degree=5)
> print(pcet)
```

```
Total number of monomials: 4367
Number of inputs:  11
Polynomial degree:  5
Number of rows: 9800
```

3 Computations

The function **calcPLSPCE** runs the computations. 35 components are required. An object of class PLSPCE is created, on which the method **print** is applied.

```
> rett  <- calcPLSPCE(pcet, nc=35)
> print(rett)
```

```
Explanation level of the response (R2, percentage and cumulated percentage)
        R2     %R2 %R2cumulated
c1  0.4612 48.2885      48.2885
c2  0.2428 25.4252      73.7137
c3  0.0927  9.7059      83.4195
c4  0.0485  5.0837      88.5032
c5  0.0273  2.8619      91.3651
c6  0.0196  2.0546      93.4197
c7  0.0125  1.3098      94.7295
```

```
c8  0.0097  1.0107      95.7402
c9  0.0073  0.7685      96.5087
c10 0.0051  0.5365      97.0452
c11 0.0043  0.4501      97.4954
c12 0.0038  0.3957      97.8910
c13 0.0034  0.3592      98.2503
c14 0.0025  0.2633      98.5136
c15 0.0021  0.2165      98.7301
c16 0.0016  0.1669      98.8970
c17 0.0014  0.1509      99.0479
c18 0.0012  0.1219      99.1698
c19 0.0011  0.1137      99.2835
c20 0.0010  0.1099      99.3934
c21 0.0008  0.0861      99.4795
c22 0.0007  0.0760      99.5555
c23 0.0006  0.0618      99.6173
c24 0.0005  0.0549      99.6721
c25 0.0005  0.0507      99.7229
c26 0.0004  0.0433      99.7661
c27 0.0004  0.0435      99.8096
c28 0.0003  0.0335      99.8431
c29 0.0003  0.0297      99.8728
c30 0.0003  0.0274      99.9001
c31 0.0002  0.0249      99.9250
c32 0.0002  0.0203      99.9453
c33 0.0002  0.0192      99.9645
c34 0.0002  0.0180      99.9825
c35 0.0002  0.0175     100.0000

Explanation-prediction level of the response (Q2 and Q2cum)
        Q2  Q2cum
c1  0.4605 0.4605
c2  0.4498 0.7032
c3  0.3118 0.7957
c4  0.2371 0.8441
c5  0.1745 0.8713
c6  0.1516 0.8908
c7  0.1133 0.9032
c8  0.0984 0.9127
c9  0.0827 0.9199
c10 0.0621 0.9249
c11 0.0552 0.9291
c12 0.0515 0.9327
c13 0.0491 0.9360
c14 0.0368 0.9384
c15 0.0307 0.9403
c16 0.0235 0.9417
c17 0.0214 0.9429
c18 0.0168 0.9439
c19 0.0159 0.9448
```

```
c20 0.0149 0.9456
c21 0.0111 0.9462
c22 0.0089 0.9467
c23 0.0062 0.9470
c24 0.0047 0.9472
c25 0.0041 0.9475
c26 0.0023 0.9476
c27 0.0023 0.9477
c28 0.0000 0.9477
c29 0.0000 0.9477
c30 0.0000 0.9477
c31 0.0000 0.9477
c32 0.0000 0.9477
c33 0.0000 0.9477
c34 0.0000 0.9477
c35 0.0000 0.9477

Optimal number of components:  28

Explanation level of the optimal number of components
       R2    %R2 %R2cumulated
c28 3e-04 0.0335      99.8431

Explanation-prediction level of the optimal number of components
    Q2  Q2cum
c28  0 0.9477

Root Mean Square Prediction of the optimal number of components
     rmsep
c28 0.2163

PLS-PCE sensivity indexes
           LE     PE    TPE
lati   0.0000 0.0068 0.1339
day    0.0000 0.0002 0.0962
xmax   0.0000 0.0002 0.0518
ymax   0.0000 0.0001 0.0549
voxel  0.0301 0.0334 0.2387
height 0.1229 0.2115 0.4474
width  0.1478 0.2048 0.4650
LA     0.0109 0.0158 0.1333
k      0.0165 0.0186 0.1299
RH50   0.0004 0.0027 0.0664
b      0.0029 0.0033 0.0766

%PLS-PCE sensivity indexes
            LE      PE     TPE
lati    0.0016  1.3608  7.0710
day     0.0041  0.0315  5.0802
xmax    0.0069  0.0409  2.7341
```

```
ymax     0.0120  0.0290  2.9002
voxel    9.0713  6.7063 12.6035
height  37.0589 42.5302 23.6198
width   44.5747 41.1830 24.5473
LA       3.2945  3.1743  7.0395
k        4.9729  3.7417  6.8596
RH50     0.1346  0.5437  3.5033
b        0.8686  0.6587  4.0415
```

The optimal number of components is 28.

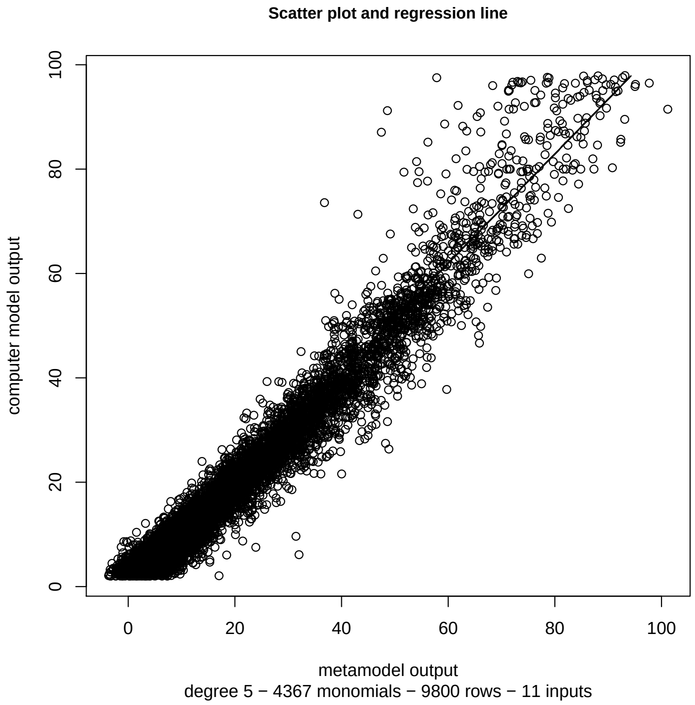
4 Plots

By default, the **plot** method draws three differents plots. Here, we draw each one after the other, by using the argument `options`.

The first argument of the **plot** method is the `PLSPCE` object created by function **calcPLSPCE**. The second argument is the `PCEpoly` object created by function **polyLeg**.

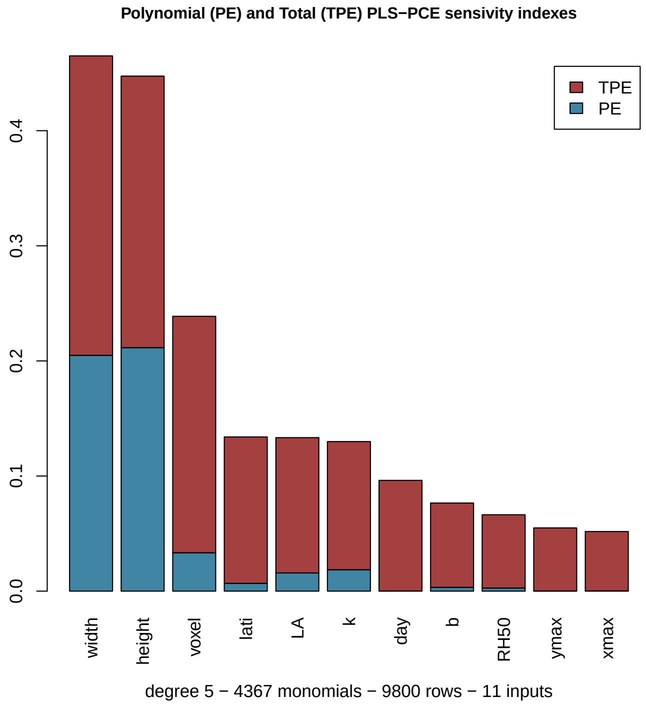
4.1 Computer model outputs against metamodel outputs

```
> plot(rett, pcet, options="fit")
```



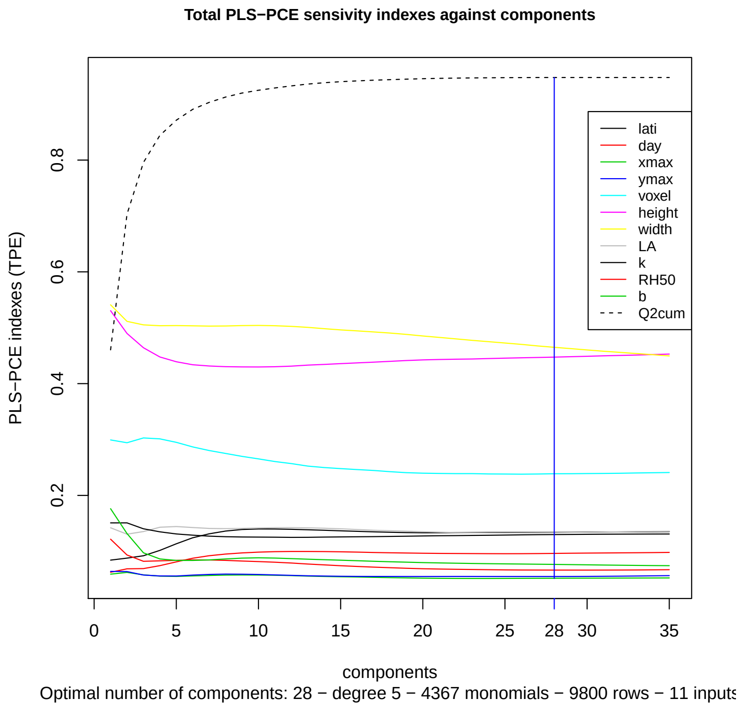
4.2 Barplots of the polynomial and total polynomial effects (PE and TPE)

```
> plot(rett, pcet, options="bar")
```



4.3 TPE against components

```
> plot(rett, pcet, options="compo")
```



5 Execution time

The CPU time required to run this example was 8.8hours on a processor with the following characteristics:

```
model name : Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
CPU MHz :          800.000
Number of cores : 4
```