

Supplementary Material

1 PCATS API

Table S1 lists the PCATS functions and provides brief descriptions of their functionalities.

R	Python	Description
<code>staticGP</code>	<code>staticgp</code>	Performs a data analysis for data with non-adaptive treatment(s).
<code>dynamicGP</code>	<code>dynamicgp</code>	Performs a data analysis for data with adaptive treatments.
<code>staticGP.cate</code>	<code>staticgp_cate</code>	Performs the calculation of the conditional average treatment effect of user-specified treatment groups for non-adaptive treatment.
<code>dynamicGP.cate</code>	<code>dynamicgp_cate</code>	Performs the calculation of the conditional average treatment effect of user-specified treatment groups for adaptive treatment.
<code>job_status</code>	<code>job_status</code>	Returns the status of a job.
<code>wait_for_result</code>	<code>wait_for_result</code>	Returns a job status until it's completed.
<code>printgp</code>	<code>printgp</code>	Prints out summary tables of results

Table S1. Functions in the PCATS.

Users can get the data, R and Python code of the examples at https://github.com/pcats-api/pcats_api_examples. The simulation datasets, R codes, and Python code used in the simulation studies are available at https://github.com/pcats-api/pcats_sim_example/.

2 INPUT PARAMETERS IN THE FUNCTIONS

- `staticGP` function in R and `staticgp` function in Python

The listed name of the input parameters are for the R function. For Python, the period (.) in the name of the input parameters should be replaced by an underscore (_).

Input parameters	Description
<code>datafile</code>	File to upload (.csv or .xls).
<code>dataref</code>	Reference to an already uploaded file.
<code>method</code>	Method to be used. “GP” for GP method and “BART” for BART method. The default value is “BART”.
<code>outcome</code>	Outcome variable.
<code>outcome.type</code>	Type of outcome. “Continuous” for continuous outcome and “Discrete” for binary outcome. The default value is “Continuous”.
<code>outcome.bound_censor</code>	“neither” if the outcome is not bounded; “bounded” if the outcome is bounded. The default value is “neither”.
<code>outcome.lb</code>	Putting a lower bound if the outcome is bounded.
<code>outcome.ub</code>	Putting an upper bound if the outcome is bounded.

outcome.link	Link function for outcome. “identity” if no transformation needed; “log” for log transformation; and “logit” for logit transformation. The default value is “identity”.
treatment	Treatment variable.
x.explanatory	Vector of the name of explanatory variables.
x.confounding	Vector of the name of confounding variables.
tr.type	Type of the treatment. “Continuous” for continuous treatment and “Discrete” for categorical treatment. The default value is “Discrete”.
tr.values	Vector of user-defined values for the calculation of ATE if the first treatment is continuous.
time	Time variable.
time.value	Pre-specified time exposure.
c.margin	Optional vector of user-defined values of c for PrTE.
tr.hte	Optional vector specifying variables which may have heterogeneous treatment effect with the treatment variable.
burn.num	Number of burn-in MCMC samples. The default value is 500.
mcmc.num	Number of MCMC samples after burn-in. The default value is 500.
x.categorical	Optional vector of the name of categorical variables in data.
mi.datafile	File to upload (.csv or .xls) that contains the imputed data.
mi.dataref	Reference to an already uploaded file that contains the imputed data.
sheet	If datafile or dataref points to an Excel file, this variable specifies which sheet to load.
mi.sheet	If mi.datafile or mi.dataurl points to an Excel file, this variable specifies which sheet to load.
seed	Random seed. The default value is 5000.

Table S2: Input parameters in the `staticGP` function of PCATS.

- `dynamicGP` function in R and `dynamicgp` function in Python

The listed name of the input parameters are for the R function. For Python, the period (.) in the name of the input parameters should be replaced by an underscore (_).

Input	Description
datafile	File to upload (.csv or .xls).
dataref	Reference to already uploaded file.

method	Method to be used. “GP” for GP method and “BART” for BART method. The default value is “BART”.
outcome	Outcome variable.
stg1.outcome	Intermediate outcome variable at stage 1
stg1.treatment	Treatment variable at stage 1
stg1.time	Time variable at stage 1.
stg1.time.value	Pre-specified time exposure at stage 1.
stg1.x.explanatory	Vector of the name of explanatory variables at stage 1.
stg1.x.confounding	Vector of the name of confounding variables at stage 1.
stg1.tr.hte	Optional vector specifying variables which may have heterogeneous treatment effect with the treatment variable at stage 1.
stg1.outcome.bound_censor	The default value is “neither”. “neither” if the intermediate outcome is not bounded. “bounded” if the intermediate outcome is bounded.
stg1.outcome.lb	Putting a lower bound if the intermediate outcome is bounded.
stg1.outcome.ub	Putting a upper bound if the intermediate outcome is bounded.
stg1.outcome.type	Type of the intermediate outcome at stage 1. “Continuous” for continuous intermediate outcome and “Discrete” for binary intermediate outcome. The default value is “Continuous”.
stg1.outcome.link	Link function for the intermediate outcome; the default value is “identity”. “identity” if no transformation needed. “log” for log transformation. “logit” for logit transformation.
stg1.tr.values	Vector of user-defined values for the calculation of ATE if the treatment variable at stage 1 is continuous.
stg1.tr.type	The type of treatment at stage 1. “Continuous” for continuous treatment and “Discrete” for categorical treatment. The default value is “Discrete”.
stg1.c.margin	An optional vector of user-defined values of c for PrTE at stage 1.
stg2.outcome	Outcome variable at stage 2
stg2.treatment	Treatment variable at stage 2
stg2.time	Time variable at stage 2.

stg2.time.value	Pre-specified time exposure at stage 2.
stg2.x.explanatory	Vector of explanatory variables at stage 2.
stg2.x.confounding	Vector of confounding variables at stage 2.
stg2.tr1.hte	At stage 2, optional vector specifying variables which may have heterogeneous treatment effect with the stage 1 treatment variable
stg2.tr2.hte	At stage 2, optional vector specifying variables which may have heterogeneous treatment effect with the stage 2 treatment variable.
stg2.outcome.bound_censor	“neither” if the outcome is not bounded, and “bounded” if the outcome is bounded. The default value is “neither”.
stg2.outcome.lb	Putting a lower bound if the outcome is bounded.
stg2.outcome.ub	Putting a upper bound if the outcome is bounded.
stg2.outcome.type	Type of outcome at stage 2. “Continuous” for continuous outcome and “Discrete” for binary outcome. The default value is “Continuous”.
stg2.outcome.link	Link function for the outcome at stage 2. “identity” if no transformation needed; “log” for log transformation; and “logit” for logit transformation. The default value is “identity”.
stg2.tr.values	Vector of user-defined values for the calculation of ATE if the treatment variable at stage 2 is continuous.
stg2.tr.type	Type of treatment at stage 2.
stg2.c.margin	Optional vector of user-defined values of c for PrTE at stage 2.
burn.num	Number of burn-in MCMC samples. The default value is 500.
mcmc.num	Number of MCMC samples after burn-in. The default value is 500.
x.categorical	Vector of categorical variables in data.
mi.datafile	File to upload (.csv or .xls) that contains the imputed data.
mi.dataref	Reference to already uploaded file that contains the imputed data.
sheet	If datafile or dataref points to an Excel file, this variable specifies which sheet to load.
mi.sheet	If mi.datafile or mi.dataref points to an Excel file, this variable specifies which sheet to load.
seed	Random seed. The default value is 5000.

Table S3: Input Parameters in the `dynamicGP` function

- `staticGP.cate` and `dynamicGP.cate` functions in R, and `staticgp_cate` and `dynamicgp_cate` functions in Python

The listed name of the input parameters are for the R function. For Python, the period (.) in the name of the input parameters should be replaced by an underscore (_).

Input parameters	Description
jobid	Job id of the computation obtained after executing <code>staticGP</code>
x	Name of variable which may have the heterogeneous treatment effect
control.tr	Value of the treatment variable as the reference group
treat.tr	Value of the treatment variable compared to the reference group
c.margin	Optional vector of user-defined values of c for the calculation of PrCTE

Table S4: Input parameters in the `staticGP.cate`, `staticGP_cate`, `dynamicGP.cate`, and `dynamicGP_cate` functions

3 R AND PYTHON CODE OF EXAMPLES

3.1 Example 1

```
library(pcatsAPIClientR)

download.file("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example1.csv", destfile
             ="example1.csv")
download.file("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example1_midata.csv",
             destfile="example1_midata.csv")

#Step 1: submit a request
jobid <- pcatsAPIClientR::staticGP(datafile="example1.csv",
                                   outcome="Jadas6",
                                   treatment="treatment_group",
                                   #specify the prognostic variables W
                                   x.explanatory="age,Female,chaq_score,RF_pos,private
                                                ,Jadas0,timediag",
                                   #specify the confounders V
                                   x.confounding="age,Jadas0,chaq_score,timediag",
                                   #add the term if you think the treatment effect is different in RF_pos group
                                   tr.hte="RF_pos",
                                   #specify the time variable,
                                   time="diffvisit",
                                   #specify the time value used to calculate the ATE
                                   time.value=180,
                                   burn.num=500, mcmc.num=500,
                                   #specify the type of outcome
                                   outcome.type="Continuous",
```

```

        method="GP",
        #specify the type of treatment
        tr.type="Discrete",
        outcome.lb=0,
        outcome.ub=40,
        outcome.bound_censor="bounded",
        x.categorical="Female,RF_pos,private",
        #specify the value c used to calculate PrTE
        c.margin="0,1",
        mi.datafile="example1_midata.csv")

#retrieve the job id
cat(paste0("JobID: ",jobid,"\n"))

#Step 2: check the request status using the job id
#If the job completed successfully, the function will return "Done".
status <- pcatsAPIclientR::wait_for_result(jobid)

#To retrieve a job status without waiting for the completion, one may use the following code.
status <- pcatsAPIclientR::job_status(jobid)

#Step 3: print the results
if (status=="Done") {
    cat(pcatsAPIclientR::printgp(jobid))
}

```

Listing 1. R code of Example 1

The same tasks could be performed by calling PCATS API from a Python shell.

```

import pcats_api_client as pcats_api
import requests

r = requests.get("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example1.csv")

with open("example1.csv", 'wb') as f:
    f.write(r.content)

r = requests.get("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example1_midata.csv")

with open("example1_midata.csv", 'wb') as f:
    f.write(r.content)

#Step 1: submit a request
jobid=pcats_api.staticgp(datafile="example1.csv",
    outcome="Jadas6",
    treatment="treatment_group",
    #specify the prognostic variables W
    x_explanatory="age,Female,chaq_score,RF_pos,private,Jadas0,timediag",
    #specify the confounders V
    x_confounder="age,Jadas0,chaq_score,timediag",
    #add the term if you think the treatment effect is different in RF_pos group
    tr_hte="RF_pos",
    #specify the time variable,
    time="diffvisit",

```

```

#specify the time value used to calculate the ATE
time_value=180,
burn_num=500,
mcmc_num=500,
#specify the type of outcome
outcome_type="Continuous",
method="GP",
#specify the type of treatment
tr_type="Discrete",
outcome_lb=0,
outcome_ub=40,
outcome_bound_censor="bounded",
x_categorical="Female,RF_pos,private",
#specify the value c used to calculate PrTE
c_margin="0,1",
mi_datafile="example1_midata.csv")

#retrieve the job id
print("JobID: {}".format(jobid))

#Step 2: check the request status using the job id
#If the job completed successfully, the function will return "Done".
status=pcats_api.wait_for_result(jobid)

#Step 3: print the results
if status=="Done":
    print(pcats_api.printgp(jobid))
else:
    print("Error")

```

Listing 2. Python code of Example 1

3.2 Example 1 - CATE

```

# CATE
jobidcate <- pcatsAPIclientR::staticGP.cate(jobid=jobid,
                                           x="RF_pos",
                                           control.tr="1",
                                           treat.tr="0",
                                           c.margin="0,1")

status <- pcatsAPIclientR::wait_for_result(jobidcate)

if (status=="Done") {
  cat(pcatsAPIclientR::printgp(jobidcate))
}

```

Listing 3. R code of Example 1 – CATE.

```

#CATE

jobid_cate=pcats_api.staticgp_cate(jobid=jobid,
                                   x="RF_pos",
                                   control_tr="1",
                                   treat_tr="0",
                                   c_margin="0,1")

```

```

print("CATE JobID: {}".format(jobid_cate))

status=pcats_api.wait_for_result(jobid_cate)

if status=="Done":
    print(pcats_api.printgp(jobid_cate))
else:
    print("Error")

```

Listing 4. Python code of Example 1 – CATE.

3.3 Example 2

```

library(pcatsAPIclientR)

download.file("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example2.csv", destfile
             ="example2.csv")

jobid <- pcatsAPIclientR::dynamicGP(datafile="example2.csv",
                                   stg1.outcome='BMI1',
                                   stg1.treatment='A0',
                                   stg1.x.explanatory="MET,Gender,BMI0,AGE,Obesity",
                                   stg1.x.confounding="BMI0,AGE",
                                   stg1.outcome.type='Continuous',
                                   stg1.time = "time1",
                                   stg1.time.value = 90,
                                   stg2.outcome='BMI2',
                                   stg2.treatment='A1',
                                   stg2.x.explanatory="MET,Gender,BMI0,AGE,Obesity,BMI1",
                                   stg2.x.confounding="BMI0,AGE,BMI1",
                                   stg2.tr2.hte="BMI1",
                                   stg2.outcome.type='Continuous',
                                   stg2.time = "time2",
                                   stg2.time.value = 180,
                                   burn.num=500,
                                   mcmc.num=500,
                                   stg1.tr.type = 'Discrete',
                                   stg2.tr.type = 'Discrete',
                                   method='BART',
                                   x.categorical="MET,Gender,Obesity")

cat(paste0("JobID: ",jobid,"\n"))

status <- pcatsAPIclientR::wait_for_result(jobid)

if (status=="Done") {
    cat(pcatsAPIclientR::printgp(jobid))
}

```

Listing 5. R code of Example 2

```

import pcats_api_client as pcats_api
import requests

r = requests.get("https://github.com/pcats-api/pcats_api_examples/raw/main/casedata/example2.csv")

```

```

with open("example2.csv", 'wb') as f:
    f.write(r.content)

jobid=pcats_api.dynamicgp(datafile="example2.csv",
    stg1_outcome="BMI1",
    stg1_treatment="A0",
    stg1_x_explanatory="MET, Gender, BMI0, AGE, Obesity",
    stg1_x_confounding="BMI0, AGE",
    stg1_outcome_type="Continuous",
    stg1_time = "time1",
    stg1_time_value = 90,
    stg2_outcome="BMI2",
    stg2_treatment="A1",
    stg2_x_explanatory="MET, Gender, BMI0, AGE, Obesity, BMI1",
    stg2_x_confounding="BMI0, AGE, BMI1",
    stg2_tr2_hte="BMI1",
    stg2_outcome_type="Continuous",
    stg2_time = "time2",
    stg2_time_value = 180,
    burn_num=500,
    mcmc_num=500,
    stg1_tr_type="Discrete",
    stg2_tr_type="Discrete",
    method="BART",
    x_categorical="MET, Gender, Obesity")

print("JobID: {}".format(jobid))

status=pcats_api.wait_for_result(jobid)

if status=="Done":
    print(pcats_api.printgp(jobid))
else:
    print("Error")

```

Listing 6. Python code of Example 2

4 SIMULATION RESULTS

	Method	N	Bias	MAE	RMSE
$Y^{(0,1)} - Y^{(0,0)}$	BART	200	-0.031	0.177	0.245
		400	-0.029	0.132	0.182
		600	-0.002	0.102	0.140
	GP	200	0.024	0.116	0.189
		400	-0.019	0.114	0.163
		600	0.010	0.093	0.148
	LTMLE	200	-0.002	0.18	0.261
		400	-0.019	0.134	0.18
		600	-0.003	0.104	0.138

$Y^{(1,0)} - Y^{(0,0)}$	BART	200	0.015	0.251	0.361
		400	0.023	0.139	0.206
		600	0.044	0.164	0.225
	GP	200	0.023	0.248	0.295
		400	0.006	0.128	0.177
		600	0.037	0.14	0.174
	LTMLE	200	0.003	0.34	0.461
		400	0	0.179	0.241
		600	0.028	0.199	0.281
$Y^{(1,1)} - Y^{(0,0)}$	BART	200	0.061	0.204	0.324
		400	0.026	0.161	0.227
		600	0.051	0.143	0.199
	GP	200	0.019	0.193	0.268
		400	0.013	0.13	0.183
		600	0.048	0.106	0.173
	LTMLE	200	0.038	0.265	0.418
		400	0.007	0.187	0.268
		600	0.037	0.167	0.247
$Y^{(1,1)} - Y^{(1,0)}$	BART	200	0.046	0.161	0.239
		400	0.003	0.094	0.154
		600	0.007	0.078	0.124
	GP	200	-0.004	0.136	0.208
		400	0.008	0.088	0.145
		600	0.011	0.078	0.135
	LTMLE	200	0.028	0.18	0.24
		400	0.012	0.098	0.154
		600	0.011	0.098	0.133
$Y^{(1,0)} - Y^{(0,1)}$	BART	200	-0.032	0.104	0.169
		400	-0.016	0.071	0.11
		600	-0.026	0.07	0.087
	GP	200	-0.018	0.106	0.162
		400	-0.006	0.073	0.107
		600	-0.021	0.067	0.085
	LTMLE	200	-0.016	0.103	0.164
		400	-0.006	0.07	0.109
		600	-0.018	0.067	0.084

Table S5: Results of ATE Estimates under the Simulation Study Setting