

Toolkits for Simulation

Huei-Ling Chen, Merck & Co., Inc., Rahway, NJ
Qian Dong, Merck & Co., Inc., Rahway, NJ

ABSTRACT

The purpose of this paper is to describe a toolkit to assist in the simulation process for users of the SAS system. In the current version of SAS, there is no clearly defined solution to trap warnings and errors associated with initial parameter settings fit into a statistical procedure such as PROC NLP. When simulation is conducted in the SAS System using a procedure like PROC NLP to find an optimization result, warnings and errors can be generated due to the conditions of how the optimization is run and are different from those warnings/errors that indicate the condition of how the program statements are executed. It is a common practice to use a remote computer to run the batch simulation job under these conditions, and it would be helpful to get a real-time email notification when the simulation batch job is completed. The toolkit described in this paper provides three key features to resolve these demands and helps create a smooth simulation process. First, it generates a summary report to go over whether there is a warning/error message associated with each iterative process of the simulation. Second, it calculates the process duration time for each iteration of the simulation. Finally, it sends an email to notify the user that the simulation, summary of warning/error messages, and the process duration times are complete.

KEYWORDS

SIMULATION, EMAIL NOTIFICATION, ERROR WARNING MESSAGE

INTRODUCTION

When using statistical procedures to conduct a simulation, often there are more than a handful of runs required in the simulation work. The initial parameter settings in the optimization work are very critical. If a run is successful, the optimized estimation result is achieved. On the other hand, sometimes the simulation experiences an unsuccessful iteration and the SAS procedure produces warning/error messages. In the case of an unsuccessful run, it is critical to know which of the parameters generated the warning/error messages. This paper provides a toolkit set for use when conducting statistical simulation studies. Three main features are included in this toolkit. First, an easy way to capture the parameter setting in question whenever there is a pre-defined warning message from the optimization run; second, a benchmark performance result of the simulation program evaluated in terms of the time for each execution run; and third, the sending of an email notification when the simulation is completed. Along with the email notification, the simulation success/failure outcome and the duration for each run is included.

CASE STUDY

The original problem occurs when the SAS procedure PROC NLP is used to analyze the data from two clinical trials to model the amount of a daily treatment drug used as a rescue medication for asthma (number of puffs used). In this analysis, a new statistical model is proposed as a robust distribution to fit the data. The probability density function for this new model is pre-specified, and it has three parameters with nonlinear inequality constraints and boundaries dependent on one another. PROC NLP, a procedure which offers a set of optimization techniques for minimizing or maximizing a continuous nonlinear function, is chosen to conduct the maximum likelihood estimates of these parameters.

As with many procedures involving nonlinear optimization processes, PROC NLP may encounter convergence problems depending on how the initial values of the parameters are chosen. If a convergence problem occurs, the maximum likelihood estimation results of the parameters will still be provided by SAS, but warning/error messages will also be produced in the log file. For a single run or a handful runs, it is possible to check the log file manually and identify the corresponding initial parameter settings that are causing the convergence problem. However, as the analysis dataset becomes larger, for example in a longitudinal analysis framework, and in our case where the endpoint of interest is analyzed by each study day, this brute force manual checking method becomes almost impossible to implement. Thus, a solution to automate the response of SAS encountering a convergence problem in a specific run and then associating the corresponding initial parameter settings along with this run is critical.

Initial resolution of this problem was to utilize the SAS automatic macro variable &SYSERR involving a return code set by SAS procedures, the DATA step, and a macro with iterations for each run of the procedure. At the end of each iteration, a check is conducted on the SAS automatic macro variable &SYSERR. Ideally, &SYSERR will change to some non-zero value when a convergence problem is encountered which will, in turn, enable the program to trap potential warning/error messages and store the parameter settings. However, further exploration of this strategy discovered that &SYSERR will always be 0 regardless of whether or not warning/error messages are generated in the log file.

The next section, Toolkit Item 1, provides a solution for automating the process of summarizing the warning/error messages and extracting the parameter settings that cause the problem.

This toolkit strategy has been successfully applied to an extensive simulation study with a wide range of parameter settings. In this simulation study, there were 45 different parameter settings and 1000 Monte Carlo repetitions for each setting. It enabled us to evaluate the performance of the proposed statistical model under investigation in terms of mean squared errors and variances, etc. Also, it was possible to conduct a sensitivity analysis to compare the difference between the estimation results generated from runs with or without warning/error messages.

TOOLKIT Item 1: SUMMARIZE THE WARNING/ERROR MESSAGE

It has been common to use the INFILE statement to read a SAS log file into a text file and retrieve the warning / error / problematic note messages from this text file. This paper applies the same technique for the simulation log checking. In addition, the initial parameters for each statistical procedure are retrieved and placed into the text file as well. The first step is to use the %put statement to record the initial parameters for each statistical procedure into the log file. After this step, the log file is saved. When reading the log file into the text file, use the data step to associate the warning/error message with each setting of initial parameters. Lastly, summarize the result and print out each initial parameter along with the warning/error messages count.

DATA STEP CONCEPT and KEY SYNTAX

1. Use %put statement to record the initial parameters for each statistical procedure in the log.

```
%put Note: param1 = &log_mean param2 = &n_puff2;
```

2. After completing the execution of the simulation program, save the log to a text file.

```
proc printto new log="&logdir.\NLP_LOGERROR.log";  
run;
```

3. Read in the %put statement line and warning/error line from the text file. Save the file to a SAS dataset.
4. Use data step to associate the warning/error message with each setting of the initial parameters.
5. Summarize the results.

```
/** Use data step to associate the warning/error message with each setting of initial parameters. */  
proc sort data=messages out=message1;  
  by logfn descending line ;  
run;  
  
data message2;  
  set message1;  
  retain id;  
  by logfn descending line ;  
  if first.logfn then id = 1;  
  else if index(upcase(message),'NOTE')>0 then id = id + 1;  
run;  
  
data note warning error;  
  set message2;  
  if index(upcase(message),'NOTE')>0 then output note;  
  if index(upcase(message),'WARNING')>0 then output warning;  
  if index(upcase(message),'ERROR')>0 then output error;  
run;  
  
/** Summarize the result.**/  
proc summary data=warning noprint nway;  
  var line;  
  class id;  
  output out=warning(drop=_type_ _freq_) n=warning;  
run;  
  
proc summary data=error noprint nway;  
  var line;  
  class id;  
  output out=error(drop=_type_ _freq_) n=error;  
run;
```

```

proc sort data=note;
  by id;
  run;

data result;
  merge note warning error;
  by id;
  if warning = . then warning=0;
  if error = . then error = 0;
  drop id line;
  run;

```

TOOLKIT Item 2: CALCULATE THE PROCESS DURATION TIME

This toolkit item will calculate the execution time of each optimization run. Before PROC NLP for each run, the SAS function, `datetime()`, is used to return the current date and time. The start date/time is stored as variable a named `START`. After executing PROC NLP for each run, the `datetime()` function is used to store the end date and time as a variable named `STOP`. The duration of this execution is then able to be calculated. The SAS function `INTCK` is used here to return the integer number of time intervals in a given time span - the time span is defined in seconds.

KEY SYNTAX

```

data start;
  start = datetime();
  call symput('start', start);
  run;

PROC ...

data stop;
  stop = datetime();
  call symput('stop', stop);
  run;

data duradd;
  merge start stop;
  dur = intck('second',start,stop);
  format start stop datetime. dur time8.;
  run;

```

TOOLKIT Item 3: EMAIL NOTIFICATION

The SAS `FILENAME` statement, with `EMAIL` device specified, can associate a file reference with an external file and then send an electronic mail via SAS. There are several email options that can be used with this `FILENAME EMAIL` statement. In this paper, the following email options are used:

```

ATTACH =
CC =
SUBJECT =

```

KEY SYNTAX

```

filename outbox email "huei-ling_chen@merck.com"
                    subject="The Summary Report for each simulation round"
                    cc = "qian_dong@merck.com"
                    attach="C:\Huei-Ling\SimulateLog\output\SimuWarn.lst"
;

data _null_;
  file outbox;
  put 'Dear Colleagues,';
  put 'Attached is the simulation output summary from the SAS program I ran
just now.';
  put;
  put 'Thanks,';

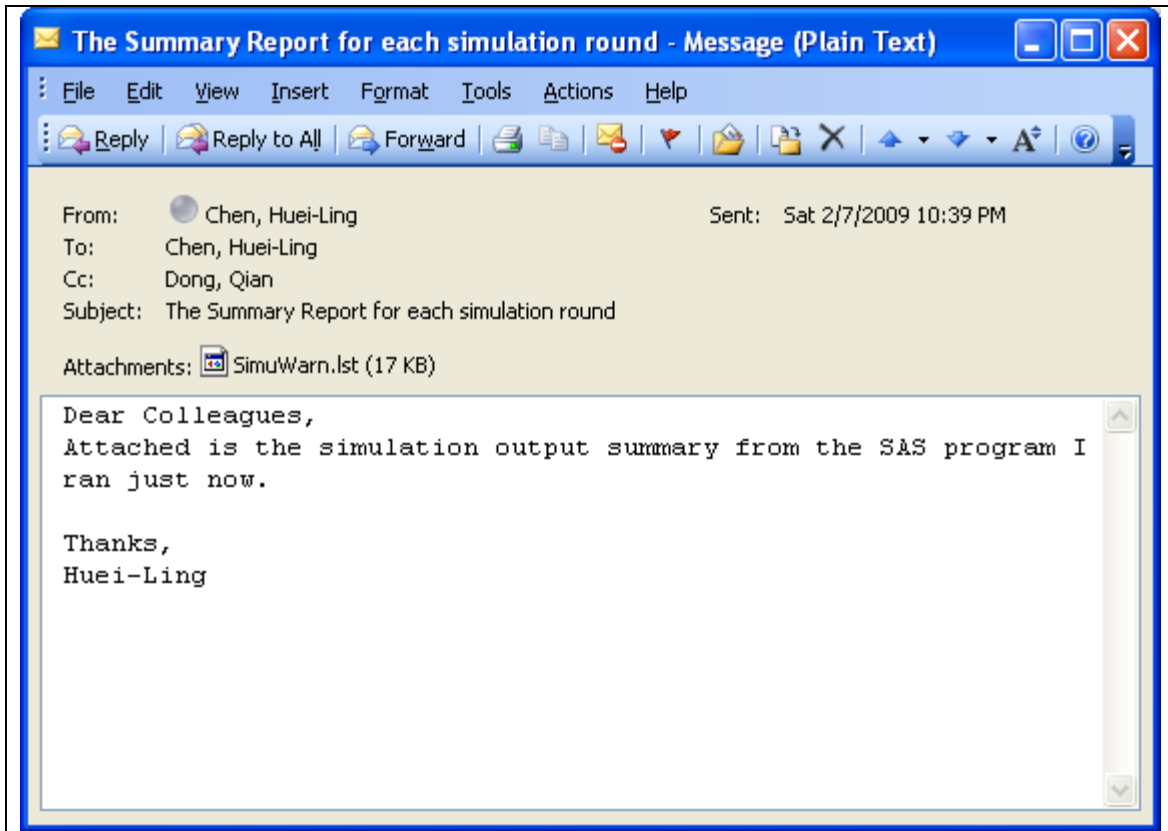
```

```
put 'Huei-Ling';  
run;
```

OUTPUT

After the simulation task is completed, the simulation log file is saved, and the simulation log file is read into a text file. The next step is to associate the warning/error message with the simulation estimation initial parameters, and the duration of execution of each simulation run calculated. At completion, the FILENAME EMAIL statement sends the output results of each of the toolkit items in an email to the recipients to notify them about the simulation result.

Snapshot of the Email:



In this paper, the simulation summary result is saved in a SAS output listing file attached in the email. In the SAS output listing file, the recipients see the parameter value, execution start date and time, execution stop date and time, and the warning or error message. This paper uses the PROC REPORT procedure to generate this summary report.

Snapshot of the Summary Report:

SimuWarn.lst - Notepad

The Summary Report for each simulation round 577
 Excutation Time and Result 22:07 Saturday, February 7, 2009

obs	param1	param2	start	stop	dur	warning	error
1	0.94277	3280	07FEB09:22:30:36	07FEB09:22:30:38	0:00:01	Y	Y
2	0.94714	830	07FEB09:22:29:05	07FEB09:22:29:06	0:00:02	Y	Y
3	0.94800	3290	07FEB09:22:30:52	07FEB09:22:30:54	0:00:01		
4	0.95408	3220	07FEB09:22:31:02	07FEB09:22:31:03	0:00:01		
5	0.95911	1920	07FEB09:22:29:48	07FEB09:22:29:49	0:00:01	Y	Y
6	0.96248	3300	07FEB09:22:30:29	07FEB09:22:30:30	0:00:01		
7	0.96684	2430	07FEB09:22:29:59	07FEB09:22:30:00	0:00:01		
8	0.96740	3280	07FEB09:22:30:44	07FEB09:22:30:46	0:00:01		
9	0.97725	2450	07FEB09:22:30:42	07FEB09:22:30:43	0:00:01		
10	0.97871	3540	07FEB09:22:29:35	07FEB09:22:29:36	0:00:02	Y	Y
11	0.98151	3650	07FEB09:22:30:31	07FEB09:22:30:32	0:00:02		
12	0.98189	2360	07FEB09:22:29:31	07FEB09:22:29:32	0:00:01	Y	Y
13	0.98278	1920	07FEB09:22:29:57	07FEB09:22:29:58	0:00:01		
14	0.98323	1040	07FEB09:22:29:09	07FEB09:22:29:10	0:00:00	Y	Y
15	0.98338	980	07FEB09:22:29:07	07FEB09:22:29:08	0:00:00		
16	0.98535	1930	07FEB09:22:30:15	07FEB09:22:30:16	0:00:01		
17	0.99008	3630	07FEB09:22:30:12	07FEB09:22:30:14	0:00:01		
18	0.99016	3600	07FEB09:22:31:04	07FEB09:22:31:05	0:00:01		
19	0.99080	2870	07FEB09:22:29:33	07FEB09:22:29:34	0:00:01		
20	0.99264	2420	07FEB09:22:30:51	07FEB09:22:30:51	0:00:01		
21	0.99401	2450	07FEB09:22:30:35	07FEB09:22:30:36	0:00:01		
22	0.99412	2990	07FEB09:22:29:17	07FEB09:22:29:18	0:00:01		
23	0.99741	3290	07FEB09:22:30:19	07FEB09:22:30:21	0:00:01		
24	1.00271	3390	07FEB09:22:29:27	07FEB09:22:29:28	0:00:01	Y	Y
25	1.00364	1950	07FEB09:22:30:06	07FEB09:22:30:07	0:00:01		
26	1.00522	2430	07FEB09:22:30:27	07FEB09:22:30:27	0:00:01		
27	1.00644	2650	07FEB09:22:29:24	07FEB09:22:29:26	0:00:01		
28	1.00680	2280	07FEB09:22:29:20	07FEB09:22:29:23	0:00:03	Y	Y
29	1.00798	3270	07FEB09:22:30:10	07FEB09:22:30:11	0:00:02		
30	1.00881	3260	07FEB09:22:30:01	07FEB09:22:30:02	0:00:01		
31	1.00911	3660	07FEB09:22:30:55	07FEB09:22:30:56	0:00:02		
32	1.01123	2430	07FEB09:22:29:40	07FEB09:22:29:41	0:00:01		
33	1.01208	1890	07FEB09:22:30:58	07FEB09:22:30:58	0:00:01		
34	1.01335	3630	07FEB09:22:29:44	07FEB09:22:29:46	0:00:02	Y	Y
35	1.01345	2450	07FEB09:22:30:17	07FEB09:22:30:18	0:00:01		

CONCLUSION

The simulation could be a lengthy iterative trial and error process. For now, there is no clearly defined solution at the end of the optimization to trap warnings and errors associated with initial parameter settings fitted into a statistical procedure such as PROC NLP. This paper provides a toolkit to associate the warning and error messages with the initial parameter settings. This method helps the statistician evaluate how the simulation performs by the different scope of the initial parameter setting. For an extensive and time-consuming simulation study, it would be helpful to get a real-time email notification when the simulation batch job is completed. Also, this paper provides a toolkit to calculate the process duration time as a benchmark performance evaluation of the simulation program. Utilizing the three toolkits, described in this paper, results in a smooth simulation batch job.

REFERENCES

Wenjie Wang and Simon Lin. 2007 "Using SAS to Send Bulk Emails with Attachments," in Proceedings of the Pharmaceutical SAS Users Group Conference (PharmaSUG 2007)

ACKNOWLEDGMENTS

The authors would like to thank the management team of Merck Research Laboratories for their advice on this paper/presentation.

Contact Information

Your comments and questions are valued and encouraged. Contact the authors at:

Huei-Ling Chen
 Merck & Co., Inc.
 126 Lincoln Avenue
 P.O. Box 2000

Rahway, NJ 07065
Phone: 732-594-2249
e-mail: Huei-Ling_Chen@merck.com

Qian Dong
Merck & Co., Inc.
126 Lincoln Avenue
P.O. Box 2000
Rahway, NJ 07065
Phone: 732-594-5993
e-mail: Qian_Dong@merck.com

TRADEMARK

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.