FAQ for SMAC version v2.06.01-master

Frank Hutter & Steve Ramage
Department of Computer Science
University of British Columbia
Vancouver, BC V6T 1Z4, Canada
`{hutter,seramage}@cs.ubc.ca`

October 19, 2013

FAQ

1 Troubleshooting Questions

1.1 My target algorithm fails to run, what should I do?

Consult the logs, and see what error messages are recorded. In some cases you may have to turn on additional options such as **--logAllProcessOutput**, and see what information is available.

    If it's not clear what the problem is consult the logs to determine the call string used (available from the logs), and try executing the target algorithm directly on the environment that it failed on. If direct execution fails, then the problem is specific to the target algorithm and/or your scenario options *i.e.* The algorithm executable cannot be found . Using the `smac-algotest` utility may make it easier to reproduce and diagnose the problem (see section 3.2) .

    If the call succeeds, the next step would be to replace the target algorithm that is being executed with a debug script/program that outputs all of the environment variables, and the call string it sees directly from

SMAC. If nothing here seems to be relevant to your problem, then you will probably want to contact us for more advice.

NOTE: If the call string that SMAC is using is not in the correct format, you probably need to use a wrapper to format it, and you should consult the *Algorithm executable / wrapper* in the manual.

NOTE: In certain Java implementations (such as Sun's Java 6) there is a bug [1] that causes the LD_LIBRARY_PATH variable to be modified internally by Java, and then the target algorithm sees the modification. In most cases this is benign, as it merely adds the location of the JVM libraries, and restates the location of the system libraries. However if library incompatibilities exist, this can cause problems. You will need to use a wrapper script to manually fix the LD_LIBRARY_PATH or upgrade Java.

1.2 SMAC gets a java.io.IOException: error=12, Cannot allocate memory when trying to execute the target algorithm

This is caused by a bug in earlier versions of Java 6 and Java 7. When SMAC executes the program in these versions, the system calls fork() and exec() which causes the entire address space footprint of SMAC to be duplicated, while this is done with traditional copy-on-write semantics, in systems which will not over commit resources, this causes the fork to fail. For more information see: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=7034935

1.3 Why does SMAC do weird things to my console (change colors, set the title, etc...)?

SMAC outputs a dump of the environment variables to console when DEBUG logging is enabled. If any of these characters are terminal escape sequences they may unintentionally cause the terminal's behaviour to change.

1.4 SMAC seems to take so much longer than tunerTimeout to execute, what can I do?

Consider using the **--wallClockLimit** option to limit the amount of wall clock time that SMAC will execute for. Additionally consider looking at the run_and_results file for the run (see *State Files* in the SMAC Manual) and see how large the difference between the recorded wall clock time and the reported runtime is.

1.5 How can I report a bug or documentation error?

Please e-mail both of the developers above.

1.6 My environment can be flaky, I don't want to ABORT but don't want to blindly continue with CRASHED

Consider using the **--abortOnCrash** option in conjunction with **--retryTargetAlgorithmRunCount**. This will allow you to retry several times before ultimately giving up.

---

[1]See http://bugs.sun.com/view_bug.do?bug_id=6670965

2 Usage Questions

2.1 System Compatibility

2.1.1 Can I run SMAC on Windows?

We have done our best to ensure that SMAC works on windows. SMAC includes `.bat` files for execution that behave similarly to the unix shell scripts. Most wrappers require that you have ruby (which is available from: ruby http://rubyinstaller.org/).

Most of the included scenarios will not work but those in the `analytic` folder and the textttsaps/saps-scenario-windows.txt should work out of the box.

2.1.2 Can I run SMAC on Mac OS X or other Unix environments?

SMAC should work on any Unix environment that supports Java 6. Unfortunately the only scenarios that are completely portable are those in the analytic folder.

2.2 I don't have any instance features, is there any benefit I can get from running SMAC?

Yes, SMAC can also run without features, and sometimes continues to show state-of-the-art performance. Additionally, we have noticed that the performance of the default configuration on all instances is a good instance feature in many situations.

2.3 Validation

2.3.1 Does SMAC support validation?

Yes, see *Offline Validation* in the manual.

2.3.2 How can I disable Validation?

Yes, by setting **--doValidation** to false.

2.3.3 Does SMAC support validating more than the last incumbent?

Yes, by setting **--validateOnlyLastIncumbent** to false. Additionally see Validation Options in the SMAC Manual appendix for options that control which incumbents can be selected.

2.3.4 How can I change the amount of memory SMAC uses?

If you are using the supplied shell scripts, you can set the SMAC_MEMORY environment variable to a positive integer corresponding to amount of memory in MB SMAC will use. If you are using some other mechanism of execution, the `-Xmx` java option is the only other way.

2.4 Parameter Configuration Space

2.4.1 Should I make a parameter categorical or numeric integral?

Numeric integral parameters are internally treated as numeric continuous parameters, and only when a call to the target algorithm is made, is the value rounded to an integer.

There is no hard and fast rule governing which method is better. If the value really is numeric and smooth, then using a numeric integral is likely to result in better predictions. If on the other hand the value is used as an index to a switch, then categorical should be better.

To illustrate using extreme examples of this:

If your value represents a bit-mask, used to control four separate options, where the most meaningful split would be based on the least significant bit (*i.e.*,the best split would be into even and odd numbers), a numeric integral value with domain [0,15] could never capture this, nor would the model ever detect it. On the other extreme, if the value really is numeric integral (*e.g.*[0, 1024]) then values that have never been seen will be randomly assigned to leafs, not capturing any smoothness.

To help inform the decision, from a SMAC internal point of view there are two places in the code where this distinction will matter. The first is during neighbourhood searches. All numeric parameters will only have 4 neighbours selected, and those neighbours will be sampled from a distribution $N(\mu, 0.2)$ where μ is the existing (*standardized* to [0, 1]) value. Any samples that fall outside of this range will be rejected.

The next distinction is within the random forest itself. Categorical values can be split arbitrarily but numeric values are always split at some constant c, with values $>$ than c going to the right, and values $\leq$ going to the left.

2.4.2 How should I encode a parameter interdependency in a PCS file (for example $0 < a < b < 10$)?

In the above case the best way to handle it, is probably to encode the parameters differently. For instance in the above, encoding the parameter b as normal (that is b [0,10] [5]i) for instance. Then for the a parameter instead encode $aScaleOfb$ [0,1.0] [0.5]i. The idea is that you simply scale the value of b between 0 and 100% to get the value of a. For example, it would take the value of b say 8, and the value of $aScaleOfb$ say 0.25 and then pass to the algorithm the decoded value of a as $8 \times 0.25 = 2$.

Using forbidden parameters in the PCS file is discouraged, as SMAC must loop over each of the settings for each configuration it generates, and it may slow SMAC down immensely. If the relation is particularly complicated, or the above doesn't work for some other reason then modifying the code to encode the relation in a more compact form is certainly feasible..

2.5 Saving and Restoring State

2.5.1 Can I continue a finished run of SMAC (*e.g.*, after a crash or running out of time)?

Yes, see *State Restoration* in the manual.

2.5.2 SMAC's state files eat up alot of disk space, can I turn them off?

Yes, set the **--stateSerializer** to *NULL*. Beginning in version 2.02, the default was to only save the files needed to restore the final set of runs, and not necessarily every iteration before hand.

2.5.3 Can I save more than just the final set of runs?

Yes, set the **--cleanOldStateOnSuccess** option to false, and a complete state will be saved on every 2^n $n \in \mathbb{N}$ iteration.

2.5.4 How can I see what iterations I can restore SMAC to?

Using the `smac-possible-restores` utility. Point it to the directory that contains the state you'd like to restore.

2.5.5 Can I change the settings of SMAC when restoring state (*e.g.,*add more instances, change objectives)?

Yes, the only thing that is restored with the state is the runs, and the state of the random objects. Everything else comes from the configuration supplied. You may want to look at the source code of `LegacyStateDeserializer` to see exactly what can be changed and what can't. NOTE: some invariants of SMAC may no longer be in force, and care must be taken. The incumbent may no longer be the best found under the new objectives, and/or the incumbent may no longer have the most number of runs [2].

2.6 How do I use the preliminary bash auto-completion with SMAC?

You can load the auto completion information by running ". ./util/bash_autocomplete.sh" at a bash prompt. For instance:

```
$. ./util/bash_autocomplete.sh
```

3 Target Algorithm & Wrapper Questions

3.1 Should I use a wrapper or modify my program to execute directly?

Either will work. It may be easier to use a wrapper but this can introduce more overhead, which (depending on the runtime of the target algorithm) may be significant (we typically see overheads up to a second). When using a wrapper it is important not to poll the output stream of the process: if the target algorithm outputs lots of data this can result in a high degree of lock contention and significantly affect the runtime performance.

3.2 How can I test that my wrapper will work with SMAC easily?

You can use the `smac-algotest` utility. It provides the same options that SMAC does with respect to executing your algorithm, but lets you specify the instance, seed & configuration. For example the following tests a random configuration of spear with a cutoff time of 50 seconds andh the `sp-var-activity-inc` variable set to *1.0924*:

```
./smac-algotest --execDir ./example_spear --paramFile ./example_spear/spear-params-mixed.txt
  --algo "ruby spear_wrapper.rb"  --cutoff_time 50  --instance
  example_data/QCP-instances/qcplin2006.10408.cnf -Psp-var-activity-inc=1.0924
--config RANDOM
```

---

[2]If you don't include the `java_obj_dump` file the incumbent should be recalculated for you. The calculated incumbent will be the one that has the most runs, and performs the best on the objective. Additionally you also lose the random object state.

3.3 What is the difference between SAT and UNSAT in the algorithm responses?

The short answer is you can use either, as SMAC makes no distinction. The reason two responses are used is a historical hold over for debugging. Essentially it was used as a cross check to ensure that some unknown bug didn't crop up, as a result of the running configuration that would break the solver. SMAC itself doesn't do anything with this information, but it's possible to use the `run_and_results` file to check the responses against the instances, and ensure that no errors were made.

3.4 What does the RANDOM Target Algorithm Evaluator do?

This Target Algorithm Evaluator basically outputs random responses, it is useful for debugging SMAC in conjunction with the **example_random** scenario.

3.5 What are the performance consequences of executing in parallel (i.e. setting --maxConcurrentAlgoExe > 1)?

It largely depends on how SMAC ends up exploring the configuration space. The largest change one could expect for a series of N algorithm executions, given infinite concurrent processors would be from $O(N)$ to $O(\lg(N))$. NOTE: This case would generally occur when we have lots of runs for the incumbent, and challengers perform very close to the incumbent's performance. If the configurations SMAC finds can be shown to perform poorly fairly quickly, then the runtime would still be about $O(N)$, and there would be no significant improvement.

Additionally when using adaptive capping with M processors, all parallel runs will be scheduled as if the other runs took 0 seconds. In other words, if we want to exploit the parallelism of running on M processors, we have to run each algorithm for M times as long, thus nullifying any advantage.

For validation however, given infinite processors the runtime would be $O(1)$, and depending on the scenario validation can be more than 50% of the runtime.

3.6 What is a Run Hash Code?

These are a mapping from the `<instance>,<seed>,<configuration setting>` to a hash. Primarily these are of use to developers, who want to ensure that a change in the code does not alter the trajectory of SMAC. Modulo the random noise in target algorithm execution, when you restore state the values should be preserved.

4 Customization and Developer Questions

4.1 There is a lack of logging options available, I need something more specific

SMAC uses slf4j (http://www.slf4j.org/), a library that allows for abstracting and replacing the logging system with ease, and uses logback (http://logback.qos.ch/) as the default logging system. While there is limited ability to change logging options via the command line (*e.g.*,**--log-level**,**--console-log-level**,**--log-all-call-strings**,**--log-all-process-output**), one can edit `conf/logback.xml`, to get much more control over the logging of SMAC. For more details of how to edit this file consult the logback documentation.

NOTE: If you replace the logger in SMAC or modify the configuration file, the logging command line options may no longer work.

4.2 I'd like to make some changes to SMAC where should I start?

The manual contains a *Developer Reference* section which should hopefully give you enough overview to start looking at the code. At the class and public method level most of SMAC's components are well documented. If you have any questions please e-mail the developers above.

4.3 I would like to optimize something that doesn't lend itself to simple command line execution, and/or would like to use something more advanced (*e.g.*, database caching, running on remote machines, etc...)

You probably need to implement a `TargetAlgorithmEvaluator`. See the *Target Algorithm Evaluator* in the manual.

4.4 I would like to use different objective functions than are provided

You will have to modify the source code, namely the `RunObjective` and `OverallObjective` classes to support the new objective. Also see Question 4.5.

4.5 Why doesn't SMAC support as many objective functions as ParamILS?

The Random Forests currently only optimize for **MEAN** and so other objective functions may have to deal with a poorer model quality. As we do not currently have other scenarios, we were unable to measure how bad the performance was and have disabled these options in the code. When **--executionMode** is set to ROAR this shouldn't be an issue.

4.6 What commits generated this version of SMAC?

Project	Version	Commit	Dirty Flag
ACLib	v2.06.01-master-648	1a58581fbab7e770e7e6994ee48ed5f93a8e376c	0
SMAC	v2.06.01-master-622	e32716a31d3b27a38810ef48906ae50620d5a0c9	0

NOTE: For non-`master` builds these commits may not contain everything in the build. (*i.e.*,non-`master` builds can be built with uncommitted changes). If the dirty flag is 0 that means the commit contains this exact copy, 1 means there were some uncommitted changes, and something else means some other error occurred when we tried to generate this.

5 Miscellaneous Questions

5.1 What are the differences between SMAC and ParamILS?

See *Differences Between SMAC and ParamILS* in the manual.

5.2 What are the relevant articles / papers about SMAC?

Several papers led up to SMAC, but the most important/up-to-date ones are (in bibtex format):

```
@InProceedings{HutHooLey11-SMAC,
  author =  {F. Hutter and H.~H. Hoos and K. Leyton-Brown},
  title =  {Sequential Model-Based Optimization for General
              Algorithm Configuration},
  booktitle = {LION-5},
  series = {LNCS},
  year =  {2011},
  pages = {507--523}
}

@InProceedings{HutHooLey11-censoring,
  author = {F. Hutter and H.~H.~Hoos and K. Leyton-Brown},
  title = {Bayesian Optimization With Censored Response Data},
  booktitle = {2011 NIPS workshop on Bayesian Optimization,
              Sequential Experimental Design, and Bandits},
  year = {2011},
Note = {Published online}
}

@INPROCEEDINGS{HutHooLey12-ParallelAC,
  author = {Frank Hutter and Holger~H. Hoos and Kevin Leyton-Brown},
  title = {Parallel Algorithm Configuration},
  booktitle = {LION-6},
  year = {2012},
  series = {LNCS},
  note = {To appear}
}
```

5.3 I'm using SMAC in academic work, what article should I cite?

Please cite the above LION-5 paper when you use SMAC. The above NIPS workshop paper is a reference for adaptive capping in the model-based framework, and the above LION-6 paper for a parallel version of SMAC.

5.4 I'd like to use SMAC for commercial purposes, what should I do?

Please contact us regarding licensing options.

5.5 What does the name after the dash in version refer to?

The name refers to the git branch, almost every public release should be *master*. Development releases are *development*.

5.6 Is the source code for SMAC available?

The source code for SMAC is included in the files `smac-src.jar`, `aclib-src.jar`, and `fastrf-src.jar`.

5.7 Can I use a different logger for SMAC?

Yes, you should be able to simply replace the logback.jars with something else that slf4j supports and it should work out of the box.