Synthesizing Configuration File Specifications with Association Rule Learning

This article is used for OOPSLA'17 artifact evaluation for the paper #199: Synthesizing Configuration File Specifications with Association Rule Learning.

1 Getting Started Guide

This artifact is in the form of a virtual machine (VM) image. This virtual machine (VM) contains the source code for the tool presented in the paper. When you login the VM, you can find all the source code and the associated paper at: `/home/tester/configErrorDetect` .

1.1 Setup Instructions

To use this tool, please first download and install a Virtual Machine manager. Any VM manager is likely to work fine, but our VM has been tested on Oracle VirtualBox 5.1.22.

After downloading and installing VMware Fusion, you could run the VM image based on the following steps:

- Extract the VM image for the artifact from the zip file (the same zip that contains this document)
- Open VMware Fusion
- Create a new virtual machine, by clicking "File → Import"
- Select the .ova file you just downloaded, and then click "continue"
- After you successfully import the VM image, the getting started phase is done. To use the VM, you can double-click it in the VM list on the left bar of your VMware Fusion. **The VM username is: tester, and the password is: oopsla**

1.2 Basic Testing

Install with `cabal install` and run the executable with `.cabal-sandbox/bin/ConfigV` to see learning and verification in action on a toy example.

The settings are preset to a testing configuration, but if you ever need to get back to the testing configuration, open `src/Settings.hs`, and set the below settings. This will make the tool learn from the files in the directory named `testLearn` and verify the files in the directory named "user". Note that both `testLearn` and `user` directories are preloaded with simple configurations to test that the tool is working correctly.

- `trainingTarget = Test`
- `verificationTarget = "user"`

To run the tool on a large example, change these settings to:

- `trainingTarget = Prob`
- `verificationTarget = "githubFiles"`

1.3 Helpful Tips

The two key files you will need to know about in order to use ConfigV are as follows:

- `src/Settings.hs`: all the useful settings to configure the tool. This file is documented in detail for each setting. NB, after changing any setting, you must `cabal install` again.
- `cachedRules.json`: the rules that were learned in json form. This can be manually inspected as a sanity check. To pretty print this file, you can use `python -m json.tool cachedRules.json`

Support and Confidence If you want to see the effect of different support and confidence thresholds, simply edit the threshold settings in the `Settings.hs` file. The values are present to the values used in the evaluation in the paper.

Your own input The best way to explore the tool at first is to keep the testing configuration and change values in the 'testLearn' files or the 'user' files. After adding more configuration settings (make up any keywords and values you

like) to 'testLearn', you can inspect cachedRules.json to directly see what was learned. You can also change the support and confidence thresholds in src/Settings.hs to control when rules will be accepted (although on the small testLearn training set this isn't very interesting).

2 Step by Step Instructions

This section describes how to reproduce the experiments from the paper (provided in the VM at /home/tester/configErrorDetect/main.pdf). By editing the `Settings.hs` file, you can reproduce any experiment from the paper.

2.1 Table 1

To generate error reports from the Github files, with probabilistic types enabled and rule graph sorting, use the following settings:

- `verificationTarget = "githubFiles"`
- `trainingTarget = Prob`
- `sortingStyle = RuleGraphDegree`
- `pROBTYPES = True`

This also produces the histogram for Fig. 5.

2.2 Table 2

To produce the errors reported in Table 2 use the following settings.

- `verificationTarget = "caseStudies"`
- `trainingTarget = Prob`

You will also need to set `pROBTYPES` to both True and False, and set `sortingStyle` to both RuleGraphDegree and Support. The ranking comes from the responses from our industry experts, saved in `home/tester/configErrorDetect/industryResponse`.

2.3 Table 3

To generate the times for learning from Table 3 use the following settings.

- `verificationTarget = "user"`
- `trainingTarget = Prob`
- `learnFileLimit = XXX`

Where XXX is set to 50, 100, 150, 200, and 256 for multiple runs. We set the verification target to "user" (which has only one file) so that the verification times are not also recorded. Timing information can be obtained with the time command or by changing line 24 in ConfigV.cabal to "-with-rtsopts= -N4 -s "