

Homework 0: Installing and Running Anglican Examples

(You don't need to submit your answer)

Hongseok Yang

KAIST

1 Introduction

This homework is almost the cut and paste of “Anglican Getting Started Guide” by Paige, van de Meent and Wood. I modified the original document slightly so as to make it fit our course. The homework will help you to run programs in the Anglican programming language and system available at the following URL:

<https://probprog.github.io/anglican/index.html>

You do not have to submit anything. However, I strongly suggest you to go through the instructions in the homework.

2 Installing Anglican

Anglican is a probabilistic programming language that compiles to Clojure which subsequently compiles to JVM bytecode. For this reason you need the Java and Clojure ecosystems installed on either your own personal computer or on a machine into which you can ssh, and, in the latter case, to which you can open socket (http) connections.

2.1 Java Prerequisites

Clojure depends on having a relatively recent Java Development Kit installed. Make sure that your machine has one. *Warning: It used to be the case that Clojure ecosystems did not work for Java 9. If you can't run leiningen, try to use OpenJDK version 8, not Java 9.*

2.2 Installing Leiningen

Leiningen is a self-installing automated Clojure project management system. You must install Leiningen from <http://leiningen.org/>. “lein” (short for Leiningen) is a self installing script as well as the primary means of invoking both Anglican and Clojure read eval print loops (REPL). Fortunately “lein” is trivial to install in *nix environments (see below). Note that Leiningen version >2.x is

required; the version in GNU-Linux package repositories may be quite a bit out of date.

The following sequence of commands will, by-and-large, install and make “lein” runnable on your system. For Unix experts the particulars are obvious and simply involve downloading and running a shell script.

```
# Download lein to /bin
mkdir /bin
cd /bin
wget https://raw.githubusercontent.com/technomancy/leiningen/stable/bin/lein

# Make executable
chmod a+x /bin/lein

# Add /bin to path
# Note: Mac OS X users should replace ‘.bashrc’ with ‘.profile’
echo 'export PATH=“$HOME/bin:$PATH”' >> /.bashrc
source /.bashrc

# Run lein
lein
```

Windows users have it just as easy. They just can use the Leiningen installer, which installs the latest version of Leiningen:

<http://leiningen-win-installer.djpowell.net/>

2.3 Cloning the Anglican-User repository

The anglican-user repository in Bitbucket contains files and templates that you can use for writing Anglican programs. Clone this repository in a local directory of a machine that you are using. To do this, move to this local directory, and run the following command:

```
git clone https://bitbucket.org/probprog/anglican-user.git
```

3 Running Anglican Programs

Anglican programs can be run as embedded code in a Gorilla REPL worksheet or as standalone programs. We recommend the former, but it is good to know how the latter also works.

3.1 Anglican Programs in Gorilla Worksheets

1. Go to the directory that contains a local copy of the anglican-user repository.
2. Run: `lein gorilla`. This produces an output that looks like:

```
Gorilla-REPL: 0.3.7-SNAPSHOT
Started nREPL server on port 53571
Running at http://127.0.0.1:53573/worksheet.html
Ctrl+C to exit
A newer version of Gorilla REPL, version 0.4.0 , is available.
```

If you do not get an automatically-loaded page in your browser, start a web browser and go to the ip address that appears in the output. For instance, for the above output, you need to write

```
http://127.0.0.1:53573/worksheet.html
```

into your browser.

3. Click the menu on the top right, select a “load a worksheet”, and load the `worksheets/template.clj` file. This file contains a template of a Anglican mode and code for performing inference on the model. You can run all the code in the template by repeatedly pressing both Shift and Enter. By the way, this template uses `predict`, which I do not recommend. The course web page will contain other worksheets that contain code from the lectures.

3.2 Anglican Programs as Standalone Programs

We suggest to have a look at the following file in the course webpage:

```
https://github.com/hongseok-yang/probprog18/blob/master/Homework/Homework0/one\_flip.clj
```

This code in the file shows how you can write a program that defines an Anglican query, runs an inference algorithm on the query, and post-processes the results of the algorithm. To run this file, do the following two.

1. Store the file in the `programs` directory of your local copy of `anglican-user`.
2. Run `lein run -m one-flip`. The `one-flip` is the namespace of all the code in the `one_flip.clj` file, and it is declared at the second line of the file.

4 Running the Regression Example from the First Lecture

The regression example from the first lecture is available at

```
https://github.com/hongseok-yang/probprog18/blob/master/Homework/Homework0/linear.clj
```

This is a Gorilla worksheet. Copy this worksheet to the `worksheets` directory of your local copy of the `anglican-user` repository. Then, load the worksheet in Gorilla, and execute all the code by pressing Shift-Enter repeatedly. Try to understand or play with the code in the worksheet.